UNCLASSIFIED

AD NUMBER
ADB016503
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; 30 JUN 1975. Other requests shall be referred to Office of the Assistant Secretary of Defense, Installations and Logistics, Pentagon (Rm 2A318), Washington, DC 20301.**
AUTHORITY
oasd ltr, 29 jul 77

THIS PAGE IS UNCLASSIFIED

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

PLEASE FOLD BACK IF NOT NEEDED FOR BIBLIOGRAPHIC PURPOSES

REPORT DOCUMENTATION PAGE

1. REPORT NUMBER APL/JHU SR 75-3D	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (*and Subtitle*) DOD WEAPON SYSTEMS SOFTWARE MANAGEMENT STUDY APPENDIX D, UNDERSEA AND LANDBASED SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED Special Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) N00017-C-72-4401
9. PERFORMING ORGANIZATION NAME & ADDRESS The Johns Hopkins University Applied Physics Laboratory Johns Hopkins Rd. Laurel, Md. 20810		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Task ZC-6
11. CONTROLLING OFFICE NAME & ADDRESS Office of the Assistant Secretary of Defense (Installations and Logistics), B. C. DeRoze Pentagon, Rm 2A318, Washington, D.C. 20301		12. REPORT DATE June 1975 13. NUMBER OF PAGES 121
14. MONITORING AGENCY NAME & ADDRESS Naval Plant Representative Office 8621 Georgia Ave. Silver Spring, Md. 20910		15. SECURITY CLASS. (*of this report*) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE NA
16. DISTRIBUTION STATEMENT (*of this Report*) Distribution of this document is limited to U.S. Government agencies because it contains Test and Evaluation Data. This study was published on 30 June 1975. Other requests for this document must be approved by the controlling office (Item 11).		
17. DISTRIBUTION STATEMENT (*of the abstract entered in Block 20, if different from Report*)		
18. SUPPLEMENTARY NOTES		

19. KEY WORDS (*Continue on reverse side if necessary and identify by block number*)

Computer Software
- Disciplined programming
- Software architecture
- Software definition
- Software deliverables
- Software design
- Software development tools
- Software engineering

(cont'd)

20. ABSTRACT (*Continue on reverse side if necessary and identify by block number*)

This appendix to the DoD Weapon Systems Software Management Study conducted by APL contains information on Undersea and Landbased Systems presented in more detail than is given in the main report. The specific systems discussed are the Trident Command and Control System, the Pershing Weapon System, and the SAM-D Weapon System. Each section is divided into a General System Description; Computer System Architecture; Computer Program Architecture; Software Definition, Design, and Implementation; Software Validation and Integration; Software Acquisition Management Organization and Methods; Operational Software Maintenance; and Highlights.

DD FORM 1 JAN 73 1473

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

19. KEY WORDS (cont'd)

Software implementation
Software integration
Software maintenance
Software management
Structured programming

Computer System Management

Milestoned development plan
Provisions for growth
Requirements analysis and validation
Resource development plan
Standard criteria for acquisition management
System integration capability
Systems engineering
Top down design

Computer Systems

Pershing Guidance and Control
Pershing Programmer Test Station
SAM-D Weapon Control Computer
Trident Central Computer Complex

Computers

AN/UYK-7
Bendix BDX 820
Burroughs D84
Raytheon WCC

Weapon Systems

Pershing
SAM-D
Trident Command and Control

APL/JHU

SR 75-3D

JUNE 1975

DOD WEAPON SYSTEMS SOFTWARE MANAGEMENT STUDY

APPENDIX D, UNDERSEA AND LANDBASED SYSTEMS

THE JOHNS HOPKINS UNIVERSITY ■ APPLIED PHYSICS LABORATORY

Johns Hopkins Road ○ Laurel, Maryland ○ 20810

Operating under Contract N00017-72-C-4401 with the Department of the Navy

ADB 016 503

ADB 016503

APL/JHU
SR 75-3D
JUNE 1975

DOD WEAPON SYSTEMS SOFTWARE MANAGEMENT STUDY

APPENDIX D, UNDERSEA AND LANDBASED SYSTEMS

Software Definition, Design & Implementation 4-18
CONF MGMT (4-33)

THE JOHNS HOPKINS UNIVERSITY ▪ APPLIED PHYSICS LABORATORY

Distribution of this document is limited to U.S. Government agencies because it contains Test and Evaluation Data. This study was published on 30 June 1975. Other requests for this document must be approved by B. DeRoze, Office of the Assistant Secretary of Defense for Installations and Logistics, The Pentagon, Washington, D. C. 20301.

ABSTRACT

This appendix to the DoD Weapon Systems Software Management Study conducted by APL contains information on Undersea and Landbased Systems presented in more detail than is given in the main report. The specific systems discussed are the Trident Command and Control System, the Pershing Weapon System, and the SAM-D Weapon System. Each section is divided into a General System Description; Computer System Architecture; Computer Program Architecture; Software Definition, Design, and Implementation; Software Validation and Integration; Software Acquisition Management Organization and Methods; Operational Software Maintenance; and Highlights.

ACKNOWLEDGMENT

The following APL staff members participated in the visits to the Undersea and Landbased Weapon Systems Program Managers, Facilities, and Contractors and/or in the preparation of the reports: R. S. Carlson, J. E. Coolahan, R. F. Gehrke, J. K. Hildabolt, R. E. Kemelhor, W. J. Sanda, T. P. Sleight, R. P. Suess, M. P. Tacey, and K. R. Wander. The editor for this Appendix was H. M. Stainer. The assistance of the Weapon Systems Program Managers and Contractors in reviewing their respective sections is gratefully acknowledged.

CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

1.1 OBJECTIVES AND APPROACH

The Weapon System phase of the APL DoD Weapon Systems Software Management Study was concerned with specific applications of software design and management to major Weapon Systems. The systems were selected to represent a variety of platforms and major missions and to illustrate all phases of the Weapon System life cycle.

The survey of individual Weapon Systems, as a major input to the overall APL study, had the following objectives:

1. To serve as a basis for understanding how and what Weapon Systems software is being or has been developed, produced, deployed, and maintained in the user environment;
2. To serve as a basis for distinguishing among the large range of uses of software in Weapon Systems; differences in function, size, and complexity; and the way these differences affect software problems and potential solutions;
3. To provide insight into the organizational relationships between the Government Program Managers, system contractors, software contractors, and Government test, maintenance, and training facilities;
4. To identify design and management techniques that have proved successful and that warrant more general application; and
5. To obtain opinions from key personnel concerning ways in which the OSD or the Services can contribute to the improvement of software cost and performance.

The survey of Weapon Systems software was carried out through the auspices of the respective Program Managers. System and software contractors were visited, where possible, to obtain first-hand information on system characteristics and development methods.

The selected Undersea and Landbased Weapon Systems are listed in Table 1-1. Two other appendices in this study discuss Airborne Systems and Shipborne Systems. These three appendices present more detailed information than was given in Section 4 of the main report.

TABLE 1-1

UNDERSEA AND LANDBASED SYSTEMS INVESTIGATED

Weapon System Programs	Systems	Status
Trident	Command and Control System	Production
Pershing	Weapon System	Deployed
SAM-D	Weapon System	Advanced Development

The individual discussions vary in detail because of the differing stages of development of the different systems. The following kinds of information were sought:

1. General System Description: A sufficient description to provide understanding of the overall system mission and requirements and the operating environment of the embedded computer system;

2. Computer System Architecture: The selection of computing equipments and their operating relationships, including the functions allocated to each computational unit;

3. Computer Program Architecture: The structure used in computer program design throughout the system, including allocation of functions to elements of the computer programs;

4. Software Definition, Design, and Implementation Methods: Techniques used in software system design management and control, especially those which have had apparent success;

5. Software Validation and Integration Methods: Management techniques, testing tools and techniques, and facilities used in software quality assurance;

6. Software Acquisition Management Organization and Methods: Methods used by the Government, system contractor, and software contractor to manage the process of software design and validation; and

7. Operational Software Maintenance: Approach used or plans for transfer of developed software to Government control for lifetime support and maintenance.

Each paragraph in the HIGHLIGHTS section for each Weapon System is followed by one or more designations [e.g., (SE1)] in parentheses. These designations indicate the APL recommendation(s) from the main report that correlate most closely with the particular highlight.

1.2 UNDERSEA AND LANDBASED SYSTEMS

One submarine system and two landbased systems were selected to complete the survey of representative software development programs.

The Trident submarine is in development as an upgrading of the current Fleet Ballistic Missile (FBM) fleet. Trident includes two major Weapon Systems, one strategic and one tactical. The tactical system, which was examined in this study, will employ the AN/UYK-7 computer that is now a standard for surface units.

The two Army landbased systems examined, Pershing and SAM-D, have particularly stringent space, weight, and power requirements because of their need for mobility. Both systems have selected computers specially tailored to their needs. Pershing selected commercially available computers; SAM-D has developed a new computer.

The Pershing system has been deployed since 1964. SAM-D is in the advanced development phase.

Table 1-2 lists the computers employed in these three systems.

TABLE 1-2

UNDERSEA AND LANDBASED COMPUTERS

Computer Designation	Word Length (bits)	Cycle Time (μs)	System	Number of CPU's
AN/UYK-7	32	1.5	Trident	4
Burroughs D84	24	4	Pershing	1
Bendix BDX 820	16	2	Pershing	1
Raytheon	24	1	SAM-D	2*

*A third CPU is planned for the future

Visits to agencies concerned with the development of these systems are listed in Table 1-3.

TABLE 1-3

WEAPON SYSTEM PROGRAM VISITS

Weapon System Program	Agency Visited	Responsibility	Date (1975)
Trident	PM-2	Program Manager	2/11
	NUSC	Certification Agent	2/18
	EB/IBM	Eng. and Integ. Agent	3/7
	NAVSEC 6172	Project Director	3/12
Pershing	Missile Command, Redstone Arsenal	Program Manager	2/8
	Martin Marietta Aerospace	System Contractor	3/4
SAM-D	Missile Command, Redstone Arsenal	Program Manager	2/8

2. TRIDENT COMMAND AND CONTROL SYSTEM

2.1 GENERAL SYSTEM DESCRIPTION

The primary mission of the Trident submarine is to host a strategic Weapon System capable of delivering Intercontinental Ballistic Missiles to selected targets and ensuring the invulnerability of that Weapon System by conducting undetected submerged patrols.

The Trident system offers significant advantages over the existing Fleet Ballistic Missile (FBM) fleet in that it responds to:

1. The growth in Antisubmarine Warfare (ASW) capabilities by permitting operation in nearly four times the ocean area of the existing FBM fleet because of a missile (C-4) range twice that of the Poseidon (C-3) equipped FBM fleet;

2. The potential unavailability of overseas bases by permitting operation and home-porting out of continental United States bases; and

3. The age of some of the existing FBM fleet. For example, the 598 Class SSBN's are approximately 15 years old.

The Trident submarine operational availability date is currently April 1979.

Although the Trident mission is strategic, the submarine is equipped with a tactical system for avoidance of encounters or, failing in that objective, for conducting a successful tactical engagement.

It is this tactical system, designated the Command and Control System (CCS), that was examined during the course of APL's DoD Software Management Study. Other ship systems such as the Strategic Weapon System (SWS) and the Strategic Navigation System (SNS) were not considered.

2.1.1 Constituent Subsystems of CCS

The Command and Control System (CCS) consists of six major subsystems that operate in the Central Computer Complex (CCC). (CCS and CCC are often referred to interchangeably in a software context):

1. Sonar,

2. Defensive Weapons (DWS),

3. Command,

4. Monitoring,

5. Ship Control, and

6. Magnetic Silencing (Future).

Table 2-1 lists the primary function(s) of each of the six major subsystems.

TABLE 2-1

CCS MAJOR SUBSYSTEMS

Subsystem	Function
Sonar	Provide intelligence for passive avoidance of contacts and detection of torpedos.
DWS	Provide localization of sonar contacts and effective delivery of weapons and counter measures.
Command	Support CO/OOD in exercise of ship command and control functions.
Monitoring	Increase survivability by identifying excessive noise sources; monitor status of selected ship subsystems.
Ship Control	Provide means of controlling ship during steering and diving, hovering and compensation.
Magnetic Silencing (Future)	Compensate for perturbations in earth's magnetic field induced by the submarine.

2.1.2 Interfaces with other Subsystems

Other ship subsystems that have hardware/software interfaces with the CCC include:

1. Exterior Communications/Integrated Radio Room (IRR),

2. Interior Communications,

3. Strategic Weapon System (SWS),

4. Strategic Navigation Subsystem (SNS), and

5. Tactical Navigation.

2.1.3 CCS Acquisition History

Requirements for an integrated CCS were first formally specified in the Submarine Performance Requirements Baseline, WBS B1C05-001C of 2 January 1972. A contract to Electric Boat (EB) to accomplish studies and tradeoffs leading to a proposed configuration was awarded in early 1972. EB subcontracted to IBM Federal Systems Division to develop studies leading to a Proposed Technical Approach (PTA). The PTA was delivered in the fall of 1972. It was subsequently revised, condensed, and reissued as a Navy document in early 1973.

A system level specification and design data document was issued in the summer of 1973, which formalized the system configuration.

Level I interface testing was conducted from August to December 1974. Level II (partial program) testing commenced in January 1975 and is currently in progress.

2.1.4 CCS System Block Diagram

Figure 2-1 shows the Command and Control System in block diagram form. The Central Computer Complex uses AN/UYK-7 CPU and Extended Memory modules.

2.2 COMPUTER SYSTEM ARCHITECTURE

The Trident CCC architecture and equipment selection was determined after numerous studies summarized in the Proposed Technical Approach document.* Primary considerations driving the selection of a centralized computer concept were reduced equipment and manning costs, minimization of required ship-board space, and the requirement for an efficient casualty capability.

2.2.1 CCC General Description

The Trident CCS general purpose computing facility consists of a hardware segment designated as the Central Computer Complex (CCC) and a supporting software segment. The hardware segment includes two AN/UYK-7 computers, each configured as symmetric multiprocessors, with

*Command & Control System Engineering & Integration, Trident Submarine Proposed Technical Approach, 1 March 1973.

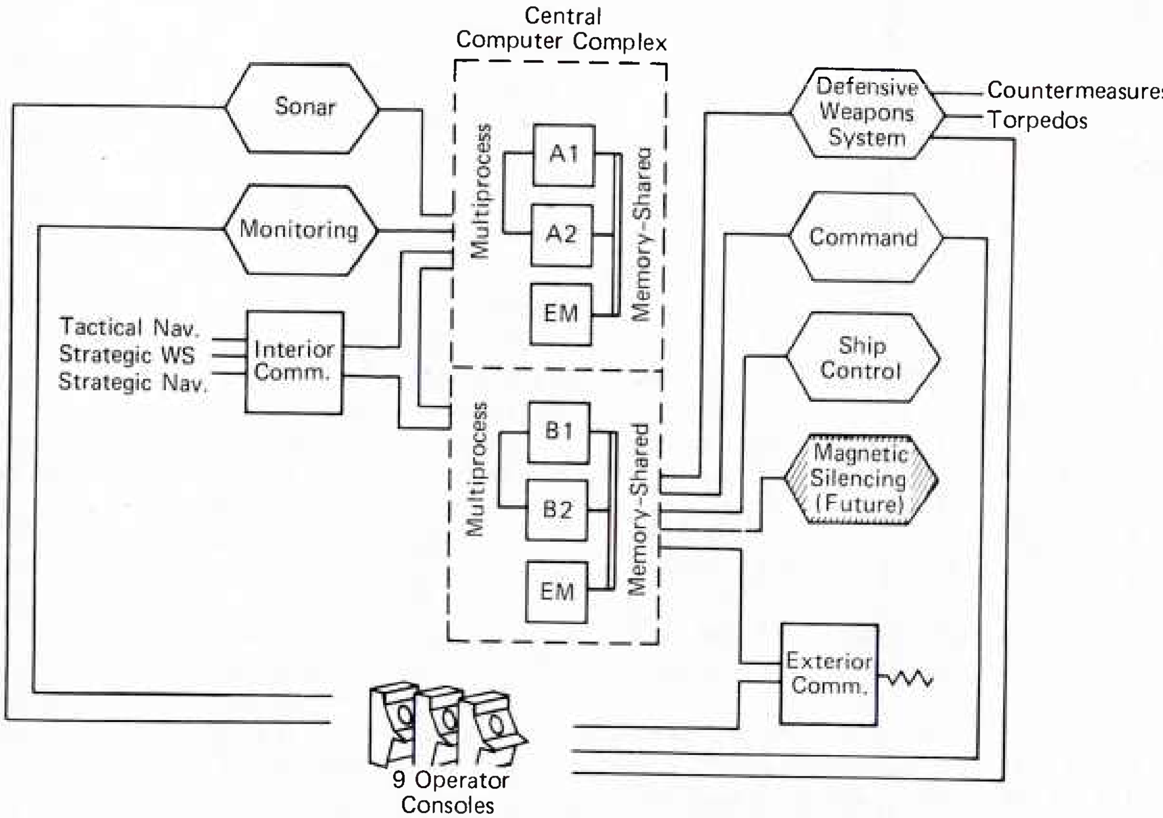


Fig. 2-1 Trident Command and Control System

peripheral equipment and interfaces to supported subsystems. These symmetric multiprocessors are designated Computer A and Computer B. Computer characteristics and operational functions are given in Table 2-2. The unit designation (e.g., A1) provides a cross reference to Fig. 2-1.

TABLE 2-2

TRIDENT CCS COMPUTER SUMMARY

Unit	Type	Function	Processor	Memory
A1, A2	AN/UYK-7 (32 bit, 1.5 μs)	Sonar data processing, system monitoring	2	180k
B1, B2	AN/UYK-7 (32 bit, 1.5 μs)	Support weapons systems, command, ship control, magnetic silencing	2	148k

The Computer A configuration is 3-2-11-2 [three frames, two central processing units (CPUs), eleven memory modules units (MMUs), two I/O controller, I/O adapters]. The Computer B configuration is 3-2-9-2 and includes two "dummy memory module units". The peripheral equipment group provides functional capabilities for on-line auxiliary mass storage of data, analog-to-digital/digital-to-analog and other conversions of input-output signals, on-line magnetic tape recording and storage of data, and manual input/hard copy printout interaction with the computers. A schematic block diagram of the CCC is given in Fig. 2-2.

2.2.2 CCC Equipment Components

The CCC hardware configuration for the CCS is two independent, symmetric AN/UYK-7 multiprocessors, interconnected by two intercomputer channels in each and configured as described below.

1. Computer A consists of 3 frames and associated power supplies, 2 CPU's, 11 MMU's, 2 input/output controllers (IOC's) with 2 input/output adapters (IOA's) of 16 channels each, 2 remote control panels and 2 maintenance panels. The interconnection of the CPU's, IOC's and MMU's is a symmetric multiprocessor, that is, each of the CPU's and IOC's are able to access all MMU's, and each of the CPU's are able to control each of the IOC's.

2. Computer B consists of 3 frames and associated power supplies, 2 CPU's, 9 MMU's, 2 IOC's with 2 IOA's of 16 channels each, 2 remote control panels, 2 maintenance panels and 2 "dummy" units for incorporation of an IOC and additional MMU's, if required. It is configured as a symmetric multiprocessor.

3. The computers are interconnected via 2 intercomputer channels that connect the respective IOC's (via IOA's) to enable data exchange between the A and B computers. The configuration provides a total of 64 I/O channels, 44 of which are assigned to specific allocations. Twenty I/O channels are unassigned, but reserved to satisfy growth requirements.

4. The peripheral equipment group (PEG) of the Central Computer Complex configuration consists of 2 mass memory RD-281(V) 3/UYK recorder/reproducers, each with 2 MX 8058/UYK disk drive units, 2 SDC's, 1 magnetic tape unit with 2 drives, and a keyboard/printer.

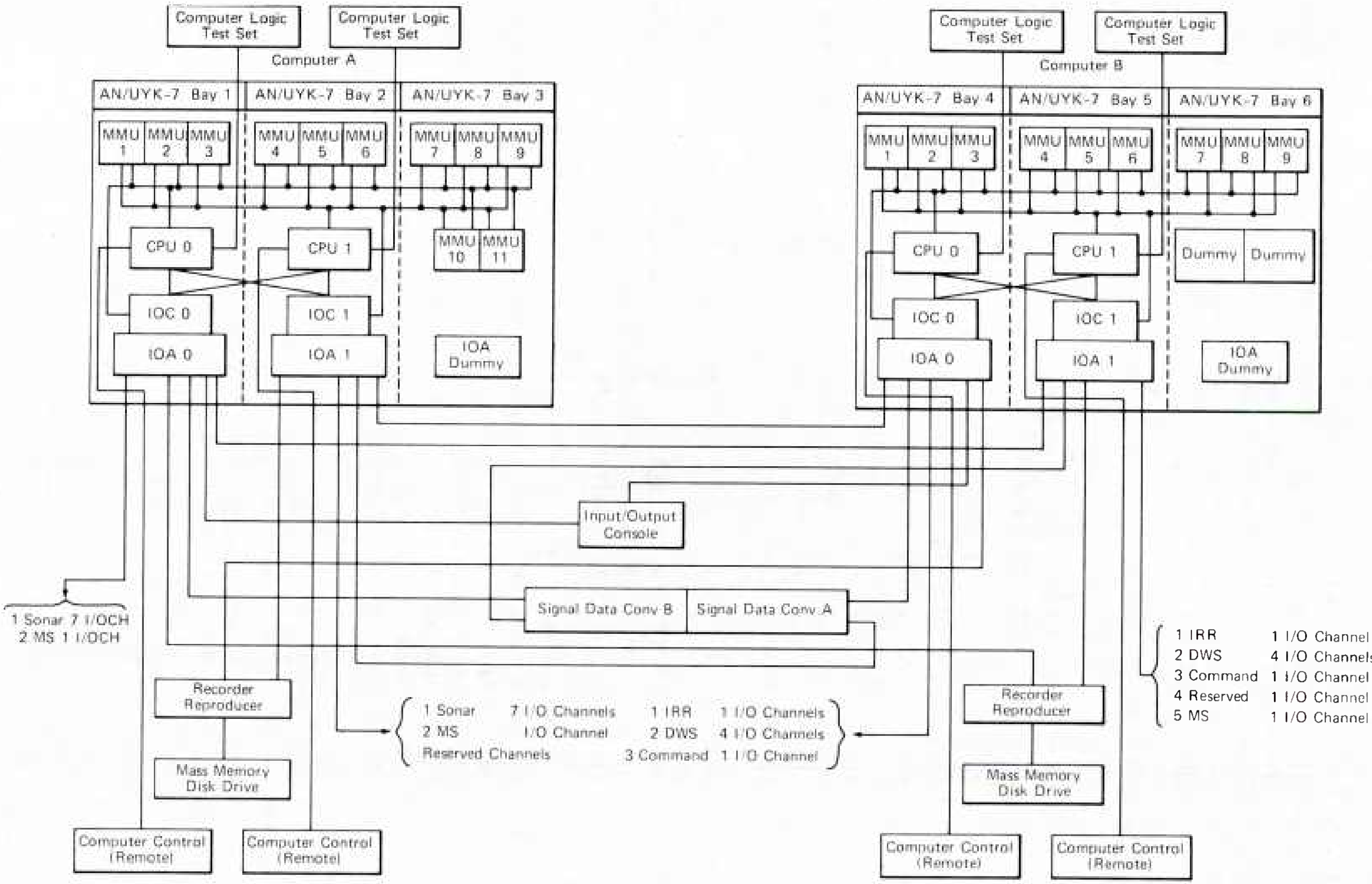


Fig. 2-2 Schematic Block Diagram of Central Computer Complex

2.3 COMPUTER PROGRAM ARCHITECTURE

2.3.1 Central Computer Complex (CCC) Software

The software system includes an advanced version of the current AN/UYK-7 Common Program (Mod X2), the Trident Service Program (TSP), Trident Application Programs, as well as AN/UYK-7 computer "stand-alone" diagnostics and fault isolation/recovery testing software.

The software components of the Central Computer Complex address the following categories of computer programs:

1. Common Program Mod X2;
2. Trident Service Program;
3. Central Computer Complex Data Base;
4. Application Programs;
5. System-level design, development, and test support software;
6. Maintenance Support Software; and
7. System-level training software.

2.3.2 CCC Program Architectural Structure

Figure 2-3 illustrates the major program elements that comprise the total software of the CCC. The figure also depicts the component modules of the Common Program Mod X2 and the Trident Service Program, which reside in both computers, and reflects the allocation of the Central Computer Complex resources with regard to subsystem application programs. Not shown in the figure are the elements for software support. Specifically, those software elements include the CMS-2Y compiler and the CMS-2Y assembler-user routines.

2.3.3 CCC Common Program Mod X2

The basic functional capabilities of the Common Program are as follows:

1. System loading,
2. System initialization,
3. System control,

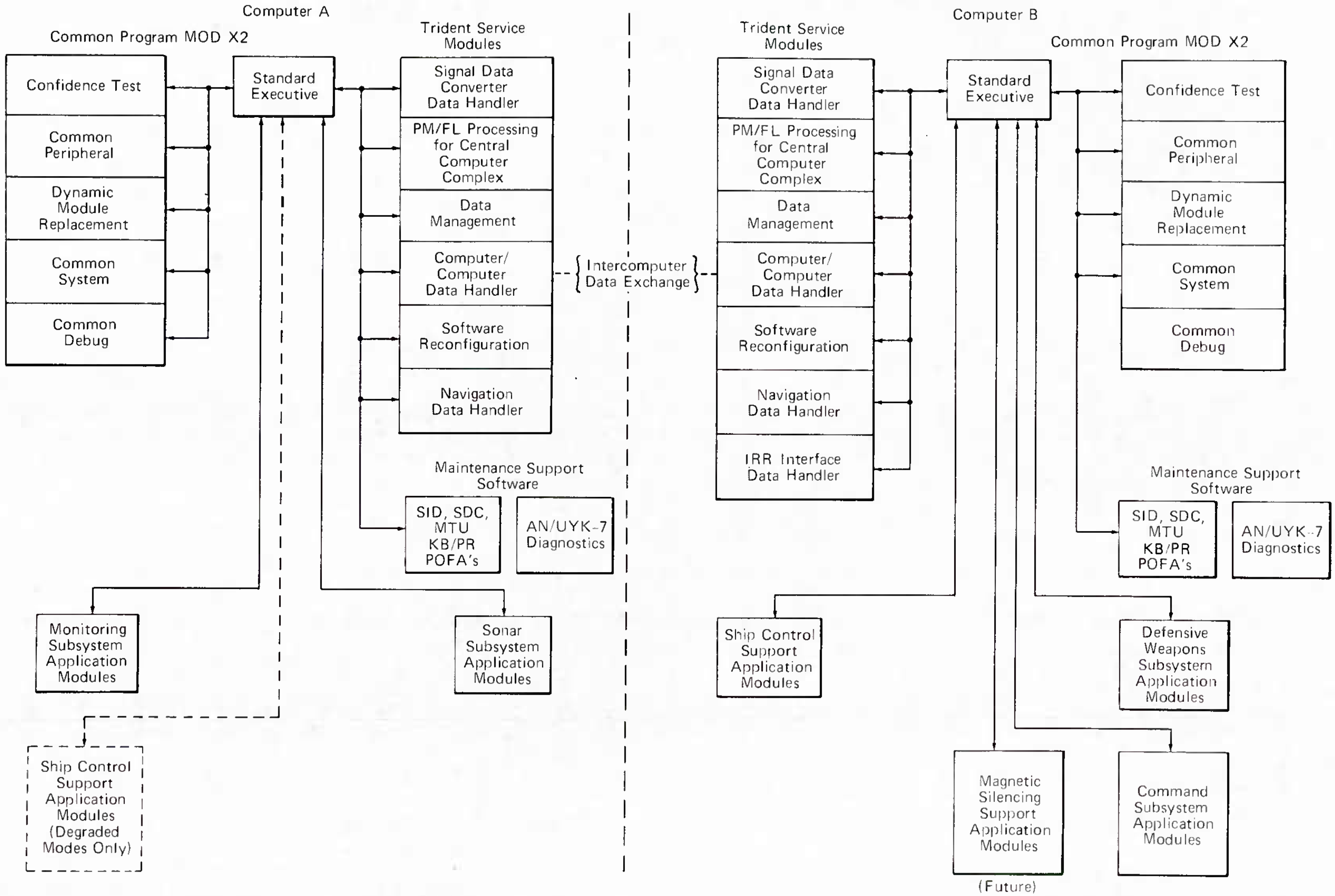


Fig. 2-3 Trident CCS Functional Block Diagram (Operational Software Segment)

4. System services,

5. Non real-time support functions,

6. Casualty recovery, and

7. Computer module fault detection

This program is sometimes referred to as the Trident Common Program (TCP).

The Common Program Mod X2 element of the CCC software segment provides the system level management and support functions normally associated with an "executive program" or "operating system" of a data processing system. The specific functions reserved for performance by the various modules of Common Program Mod X2 are described in the following subsections.

2.3.3.1 Standard Executive

This module performs the executive functions of real-time scheduling, interrupt control, centralized control of I/O processing, and control of message data transfers between application program modules of the same or different subsystems. Its scheduling algorithm accommodates the priority/precedence of data processing support functions for CCS. This module provides centralized initiation of I/O functions; however, the programming, management and control of the actual I/O operation will be a function of the user application program.

2.3.3.2 Common Peripheral

This module provides all required I/O functions for the disk, MTU, and keyboard/printer peripheral equipments of the Central Computer Complex to include device code translations, data management for auxiliary storage and retrieval, and functions associated with operator keyboard interaction with the system for real-time monitoring and control of the computers.

2.3.3.3 Common System

This module of Common Program Mod X2 provides the actual interface between application programs and common data such as common user mathematical and/or conversion routines. This module also contains hardware and software fault detection and recovery routines including a Memory/IOC Test routine for fault detection of computer components.

2.3.3.4 Dynamic Module Replacement (DMR)

This Common Program Mod X2 module includes the capability to operate in real-time to load instructions and data from mass storage into the memory area of a subsystem user application program. At the time of system initialization, this module, in addition to loading, also initiates the loaded modules for execution under the Standard Executive element of Common Program Mod X2.

2.3.3.5 Confidence Test

This module of Common Program Mod X2 provides the capability to perform "go/no-go" confidence checks on the CPU components of Computers A and B. It operates in real-time without operator intervention. It communicates a malfunction indication to error recovery routines of the Common System module for initiation of corrective action.

2.3.3.6 Debug

This module assists in real-time debugging of other program modules. It is used for program checkout during the development and testing of user application programs. It provides real-time access to the contents of arithmetic registers, memory registers and inter-module messages.

2.3.4 CCC Trident Service Program (TSP)

The Trident Service Program augments the Common Program Mod X2 services and support functions by providing modules for functions peculiar to the Trident CCS environment as listed below:

1. Signal Data Converter (SDC) Data Handler,
2. Integrated Radio Room (IRR) Data Handler,
3. Performance Monitoring/Fault Location (PM/FL) Processing for Central Computer Complex Equipment,
4. Navigation Data Handler,
5. Software Reconfiguration,
6. Data Management, and
7. Computer/Computer Data Handler.

The TSP element of the software segment provides system-level services and support, supplementary to those of Common Program Mod X2, as specifically required for the Trident CCS configuration and operation. All modules of the TSP conform to the interface standards and functional requirements of Common Program Mod X2. They reside in Computers A and B as depicted in Fig. 2-3. Major TSP modules are described in the following subsections.

2.3.4.1 Signal Data Converter (SDC) Data Handler

This module provides the capability by either Computer A or B to control either SDC-A or SDC-B, receive, process, and store data transmitted through the SDC, and send user data to devices attached to the SDC.

2.3.4.2 Integrated Radio Room (IRR) Data Handler

This module provides interface control and data formatting functions for data exchange between the IRR and the CCC.

2.3.4.3 Data Management

This module provides centralized file and data management services for multiple subsystem accesses. It provides for collection and storage of system history data, sensor readings, and status data, and retrieves data requested by various program modules. The Data Management module also provides for storage and retrieval of data on both disk and tape.

2.3.4.4 Performance Monitoring/Fault Location (PM/FL) Processing for the CCC

This module provides scheduling on a predefined periodic basis for exercise of Common System Processor Arithmetic Test for each of the CPU elements of Computers A and B. In addition, it functions as the data manager for use of confidence test and diagnostic routines of the equipments making up the Peripheral Equipment Group of the CCC.

2.3.4.5 Software Reconfiguration

This module will function when a CCC failure occurs. The module shall: (1) determine the level of system operating capability with respect to the predefined levels of degraded modes of operation, and (2) automatically initiate the appropriate system reconfiguration actions to attain the maximum level of system capability with remaining resources. This module is disk resident with the exception of a core resident routine which initiates appropriate action to transfer supporting software of Ship Control Support to Computer A in the event of failures in the primary support computer.

the application program of the subsystem, which may also include a program module for internal control and management of the task type subprograms. The application program along with related data uniquely required by that subsystem is structured and designated as a Region. Major Subsystem Application Programs are given in Fig. 2-1.

Within Computer A and Computer B, a predetermined size and area of core memory is allocated for the dedicated functional usage specified for each supported subsystem. Regions are allocated within the MMU's available to each computer with the aim of obtaining processing overlap efficiency and minimizing memory access contention.

2.3.7 Central Computer Complex Modes

The CCC operates in normal and degraded modes, defined in terms of CCS functions to be supported and the hardware/software resources available for providing the required support. Automatic reconfiguring to a degraded mode is a feature of the casualty operation. Computers A and B individually operate at the following levels of system support capability:

1. Normal (fully operational) mode of operation,
2. Bypass Configuration,
3. Degraded Configuration, and
4. Failed Configuration.

In addition, both computers will retain, in all modes of operation, a capability for detecting and supporting recovery from faults in their own and each other's operation. Peripheral group equipment of the CCC will operate in normal mode and in degraded modes. Table 2-3 illustrates the CCC normal mode resource allocation.

TABLE 2-3

CENTRAL COMPUTER COMPLEX NORMAL MODE RESOURCE ALLOCATION

Computer A	Loading % of CPU	1000's Words Core Memory
Sonar	143	103
Monitoring	16	35
TCP/TSP	21	26
Reserve	20	16
Total	200	180
Computer B	**Loading % of CPU**	**1000's Words Core Memory**
DWS	80	39
Command	20	24
Ship Control	15	18
Magnetic Silencing (Future)	10	8
TCP/TSP	22	27
Reserve	53	31
Total	200	147

2.4 SOFTWARE DEFINITION, DESIGN, AND IMPLEMENTATION

2.4.1 Command and Control System Software Definition

The basic software definition documents are shown in Table 2-4. Figure 2-4 shows the primary definition (specification) documents and their relationships with management and design documents.

TABLE 2-4

CCC SOFTWARE DEFINITION DOCUMENTS

Document	Title
NAVSHIPS 0905-497-1010	System Operational Specification for Trident Submarine CCS, 15 July 1973
NAVSHIPS 0967-029-5122	Program Specification for Common Program MOD X2, 15 January 1973
NAVSHIPS 0967-029-5152	Computer Program Performance Specification, Common Program MOD X2 Standard Executive (Preliminary) 1 August 1973
NAVSHIPS 0967-029-5154	Computer Program Performance Specification, Common Program MOD X2 Common Peripheral Module (Preliminary) 1 August 1973
NAVSHIPS 0967-029-5155	Computer Program Performance Specification, Common Program MOD X2 Dynamic Module Replacement Module (Preliminary) 1 August 1973
NAVSHIPS 0900-074-4010	Trident Command and Control System Specification (latest date of issue)
NAVSHIPS 0900-076-4010	Interface Design Specifications (IDS) Functional Interface Definitions, 14 December 1973

The CCS Contractor Electric Boat (EB) (also referred to as the Software Integrator) is responsible for project coordination and technical support, and for the technical management and performance of activities required for the design, engineering, provisioning and development (of designated software), integration test and evaluation, and demonstration of the integrated CCS software as directed and approved by the cognizant Navy agencies. EB has subcontracted with IBM to assist in carrying out these responsibilities. CCS Contractor software responsibilities include software specification.

2.4.2 CCS Software Design

The CCS Contractor's responsibilities also extend to software design. The Trident Submarine CCS Software Design Working Group (SDWG) also acts in an advisory capacity in software design matters (among others). (The SDWG will be described further in the Management section 2.6).

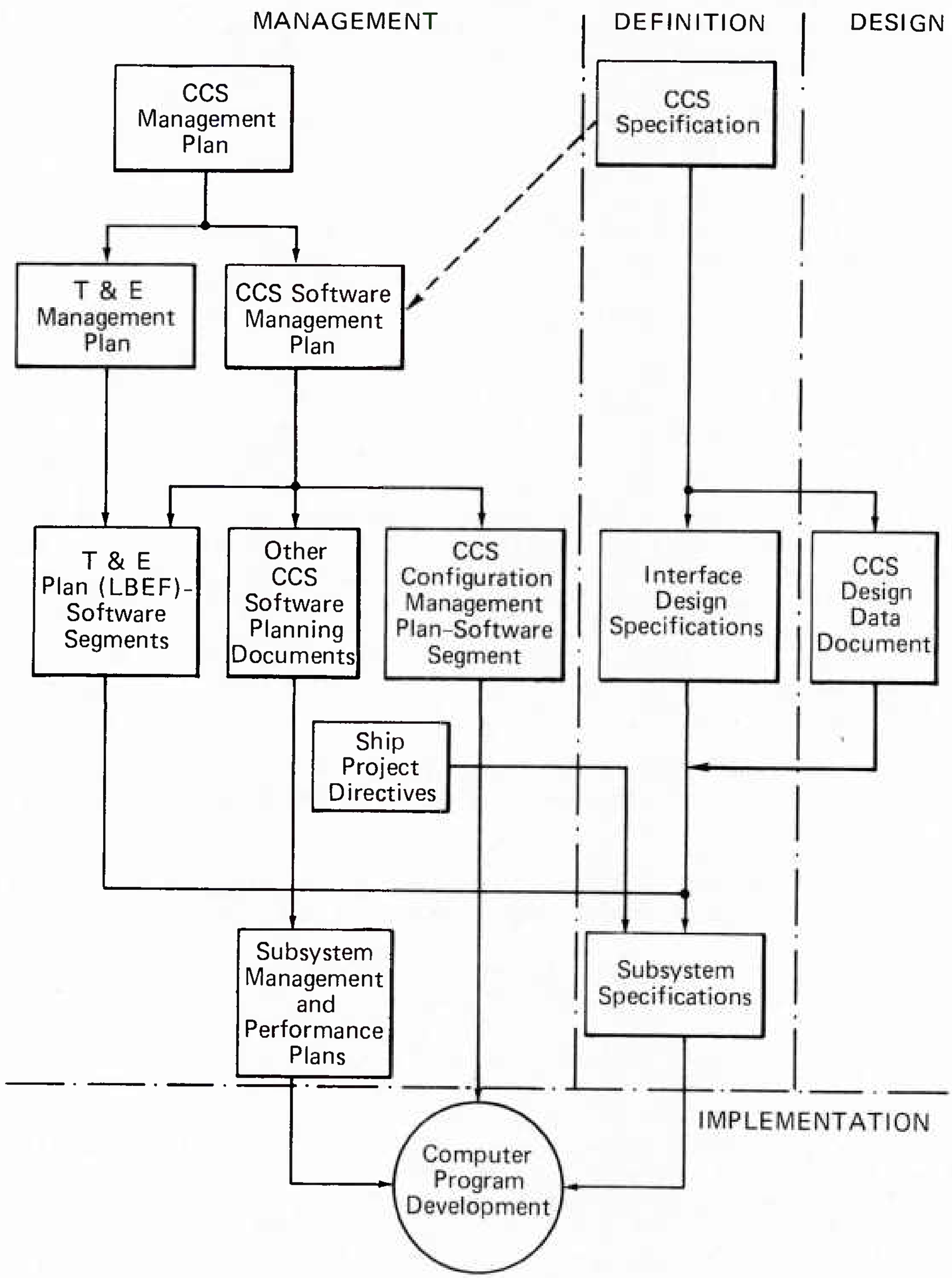


Fig. 2-4 Primary CCS Management and Specification Documents

The primary design document is NAVSHIPS 0900-075-7040, Command and Control System Trident Submarine Design Data Document, Volume IV, Part 1, Software Design and Interface Description, 31 August 1973. Other supporting documents included the Function Operational Design Documents and the Computer Subprogram Design Documents.

Design aspects are key elements of the overall CCS Software Development Plan (see Section 2.6).

Central Computer Complex Design is covered in the Planning phase, and Program Analysis and Program Design is discussed in the Computer Program Generation phase.

2.4.3 CCS Software Implementation

The Trident Submarine CCS Computer Program development is a diversified activity, and many individual agencies are required to provide its software components. To minimize the number of development site facilities, primary facilities dedicated for program compiling and checkout sufficient to support the compiling and checkout requirements of a number of the subsystems have been made available at the Land Base Evaluation Facility (LBEF). Two suites of CCC equipment will be provided at the LBEF to enable the CCS software integration and CCS system (hardware/software) integration (see Section 2.5).

The following non real-time functions facilitate the development of the Trident software system in accordance with the Trident CCS Software Management Plan, NAVSHIPS 0900-075-4010:

1. Assembler: A CMS-2Y assembler to aid programmers in developing top-down structured assembly language software code;

2. Compiler: A CMS-2Y compiler to aid programmers in developing top-down structured compiler language code;

3. User routines: Library management routines associated with the use of the CMS-2Y compiler/assembler; and

4. Computer Models: Computer models and related reports on their utilization and results.

At the LBEF Central Compile Facility (CCF), batch compiling takes place in a closed shop. An operator is on duty with equipment dedicated to supporting the compiling, loading, library, and utility functions associated with the CMS-2Y system. In addition, a time sharing system (SHARE-7) has been provided utilizing UNISCOPE-100 terminals.

2.4.3.1 CCS Implementation Aids

Top-down design and structured programming are specified for use by all Trident software developers in the Trident Software Management plan. Benefits include:

1. Early specification of milestones;

2. Improved management visibility and control;

3. Early testing of important code via phased deliveries (see Section 2.5) and the resultant detection of errors in the development phase rather than in the operational phase; and

4. A readable standard for programs and modules.

The Computer Program Generation phase of the Software Development Plan thus defines top-down structured program generation as one of its key elements.

The formal CCS Software Management Plan (Section 2.6) includes a description of the following items:

1. The development approach, particularly in the activities associated with computer program generation and integration;

2. A software development plan segmented into five overlapping phases including the schedules established for development activities and deliverable items;

3. Provisioning of computer program development and checkout facilities; and

4. Software development standards and constraints that are imposed on all computer program developers to ensure adequate means of achieving management visibility and controls required for integration, including provision for standard program documentation and development status reports.

NAVSHIPS 0900-076-5010, "Trident CCS Software Standards and Conventions" (latest date of issue) thereby aids in satisfying part of the item 4 requirements.

Finally, assistance is provided during the implementation phase by the Hardware and Support Software Agent (HSSA). HSSA is NAVSEC 6178D and has two principal responsibilities:

1. To provide computer hardware including peripherals for CCS development sites, and

2. To provide Common Program Mod X2 and development software for CCS computer programs.

2.5 SOFTWARE VALIDATION AND INTEGRATION

Software integration and testing is one of the primary areas of responsibility for the CCS contractors (EB and IBM).

The formal CCS Software Management Plan requires computer programs for the support of integration test and evaluation of CCS system hardware and software.

The CCS Software Development Plan designates several aspects of validation and integration as key elements. Thus, in Phase I - Planning, a key element is "Integration Test and Evaluation Plans." Similarly, in Phase II - Computer Program Generation, "Delivery Requirements for Integration" specifies intermediate deliveries of Level 1*, Level 2*, partial programs, and complete programs.

In Phase III - Computer Program Integration Test and Evaluation, the qualified subsystem programs will be the primary software elements to be integrated at the LBEF. During this phase, the verified software will next be subject to system-level (hardware/software) tests and then to LBEF certification performed by NUSC/Newport (as an agent of PMS-396).

In Phase IV - Shipboard Installation and Acceptance, the shipboard installation of the CCS Computer Program will commence following successful completion of integration and verification tests at LBEF. Furthermore, continued shipboard software integration support will be provided by the CCS contractors during this phase.

2.5.1 Land Based Evaluation Facility (LBEF)

The LBEF will contain two CCC hardware sets. A Resident set of CCC equipment for dedicated use in CCS software checkout, integration, and verification will be provided. It is intended that this set later be configured with subsystem hardware and remain at the LBEF for maintenance and training purposes. A second set, the Transient set, will be delivered for combat systems integration test and evaluation, including final test and demonstration of the CCC software. It will then be shipped to the first Trident Submarine. All subsequent Transient sets

*See subsection 2.5.4 for a description of the levels.

will follow the same procedures at the LBEF prior to shipboard installation. Each CCC will contain the following equipment set:

1. One AN/UYK-7 computer (3-2-9-2 configuration)

2. One AN/UYK-7 computer (3-2-11-2 configuration)

3. Two disk systems

4. One magnetic tape (two transports)

5. One keyboard printer

6. Two SDC's (A/B)

7. Three CDC Display Consoles

8. Four SID Consoles

   (a) One OJ-326(V)2/UYK Monitoring Console
   (b) Two OJ-326(V)3/UYK DWS Consoles
   (c) One OJ-326(V)4/UYK Command Console

2.5.2 Auxiliary Development Sites

A second compiling and checkout site has been established at Manassas, Virginia, to support the requirements of the Trident Submarine Sonar development. Additional users may use the facility consistent with the resources available. The hardware supplied includes:

1. One AN/UYK-7 computer (3-2-11-2 configuration);

2. One RD-281 disk system (with four drives);

3. Two UNIVAC 1840N magnetic tape (4 transports each); and

4. One 1532 console

Another small compiling facility has been established to support Sensor Interface Data System (SIDS) Software development at Hughes, Fullerton, California.

2.5.3 Test Support Programs

The following is provided for use at the Land Based Evaluation Facility (LBEF) and/or on board Trident as appropriate:

1. A Combat System Interface Test (CSIT) as specified by "Program Specification for Combat System Interface Test";

2. A Combat System Alignment Test (CSAT) as specified by "Program Specification for Combat System Alignment Test";

3. A Combat System Operability Test (CSOT) as specified by "Program Specification for Combat System Operability Test";

4. Simulation programs as specified by the "Program Specification for Simulation" (i.e., a large software system (SIM/STIM-Simulate/Stimulate) are available to support test and integration of the CCS. In particular, simulation provides a capability to induce test parameters and test scenarios, as well as to simulate hardware/software interfaces which may be lacking during the test);

5. Data Recording and Reduction Program Package as specified by "Program Specification for Data Recording and Reduction";

6. Utility Package (U-PAK) NAVSHIPS 0967-027-5270; and

7. Resource Simulation Routines for Level 1 Deliveries.*

2.5.4 Integration Approach

Trident employs phased, milestoned deliveries as both a management and a technical tool. The phased deliveries are identified as follows:

1. Level 1: Actual coding of all entrances to each subsystem's modules, with stubs used to simulate CCC resource use.

2. Level 2: Additions to the Level 1 delivery with sufficient code provided to allow interface testing between all subsystems. Stubs are again provided where coding is incomplete.

3. Partial: Complete coding of certain modules (functions) of a subsystem, with stubs also provided for those portions that are incomplete.

4. Final: Complete and certified coding of all modules.

---

*See subsection 2.5.4

These phased deliveries permit early identification of interface (integration) problems between subsystem programs and between the subsystem programs and the operating system. Results of the integration of the phased deliveries are promulgated to subsystem developers through the CCS Software Status Reports. These milestoned phased deliveries, along with the Software Status Reports, form the prime progress monitoring techniques employed.

Integration testing will focus on three major areas: (1) CCC resource utilization; (2) Normal mode for functional operation of subsystems without degradation; and (3) Degraded modes.

1. The Central Computer Complex resource utilization testing will verify that each subsystem program correctly interfaces with the equipment and common software and does not unduly interfere with the use of those resources allocated to another program. For the peripheral equipment, tests will demonstrate proper initial program loading from the disk and from magnetic tape, loading of transient program segments, data storage/retrieval via the disk, operator entries, and program printouts. For the computer hardware, tests will confirm that processor time, memory, and I/O controller time usages are within the allocated values. For shared applications, tests will verify that each subsystem program correctly performs the common functions and accesses the common data.

2. Normal mode for functional operation of each subsystem program will be verified through the use of system tactical and environmental simulation drivers called scenarios. Extraction of functional data during the performance of scenarios and the comparison of that data to subsystem performance requirements will be the method used to verify that subsystem functional degradation has not occurred as a result of integration. The content of the scenarios will be used to generate data to drive simulators representative of the environment and subsystem hardware.

3. Another area of testing is that of degraded modes. The CCS operating system will have recovery logic to handle Central Computer Complex hardware failures. The CCS system design is based on the premise that system reliability and effectiveness is such that casualties beyond first-level failures will receive minimal consideration. Testing in this area will be accomplished by simulating or generating all first-level failures and verifying that appropriate recoveries are made.

The integration approach, as described in this section, is expected to enhance Trident Quality Assurance by permitting early identification of interface problems among subsystem programs and between subsystem programs and the operating system.

2.5.5 Certification Testing

An independent agent, NUSC/NPT, has been chartered to perform an LBEF certification on the Trident hardware/software system. It is planned that approximately six 100-hour scenarios will be run at the LBEF to insure the hardware/software system will support the operational deployment of a Trident SSBN.

2.5.6 Supplemental Documentation

Supplemental documents used to support this phase of Trident development were:

1. Computer Program Test Plan and

2. Computer Program Test Procedures

2.6 SOFTWARE ACQUISITION MANAGEMENT ORGANIZATION AND METHODS

2.6.1 General Management Information

Trident management information is summarized in Table 2-5.

The responsibilities for management, acquisition, generation, test, integration, and maintenance of the CCS computer programs have been assigned to specific Navy agencies and their respective contractors. A Project Manager (PM-2) has the overall responsibility for developing the Trident Submarine. To support development and production, a Ship Acquisition Project has been established under Commander, Naval Ship System Command (PMS-396), who is designated Ship Acquisition Project Manager (SHAPM). The SHAPM has authority for issuing all Ship Project Directives for CCS activities. The Navy has contracted with General Dynamics/Electric Boat Division (EB) for the engineering and integration (E&I) of the Trident Submarine CCS. As the CCS Contractor, EB has subcontracted to the International Business Machines (IBM) Corporation the engineering required for both the hardware and the software integration efforts. The Naval Ship Engineering Center (NAVSEC) is providing design and engineering support to the SHAPM.

Primary responsibility for CCS development resides in PMS-396, supported by the CCS Project Director, NAVSEC 6182, who provides project

TABLE 2-5

TRIDENT MANAGEMENT INFORMATION

Program Manager	PM-2
System Contractor/Integrator	EM/IBM
Type Contract (System and Integrator)	Cost plus fixed fee
Program Status	Production
Software Contractors	
Sonar	IBM
Defensive Weapons/Command	NUSC
Ship Control	EB
Common/Service Programs	Univac
Maintenance Agent	Trident Maintenance Agency
Software Deliverables	As per WS 8506; plus Operating procedures Source and object library Memory map Compiler and loader card decks Compile listing tape
Certification Agent	NUSC

direction and coordination, and the CCS Contractors EB/IBM who provide systems engineering, perform integration test and evaluation, and develop technical and support plans and controls. Figure 2-5 depicts these agencies along with associated Trident Submarine committees, the CCS Steering Group and CCS Design Working Group (DWG), which are chaired by the SHAPM and CCS Project Director, respectively, and the major Participating Managers (PARM's) and their contractors. To maintain the integrity of the CCS and the CCS Software Baselines, provision was made in the CCS Configuration Management Plan for a Configuration Control Board (CCB) and a Software Configuration Control Board (SCCB).

2.6.2 Management Organizations

2.6.2.1 Computer Software Systems Manager (CSSM)

The Trident Submarine CSSM is NAVSEC 6172B4. The CSSM has been assigned responsibilities in two main areas by PMS-396:

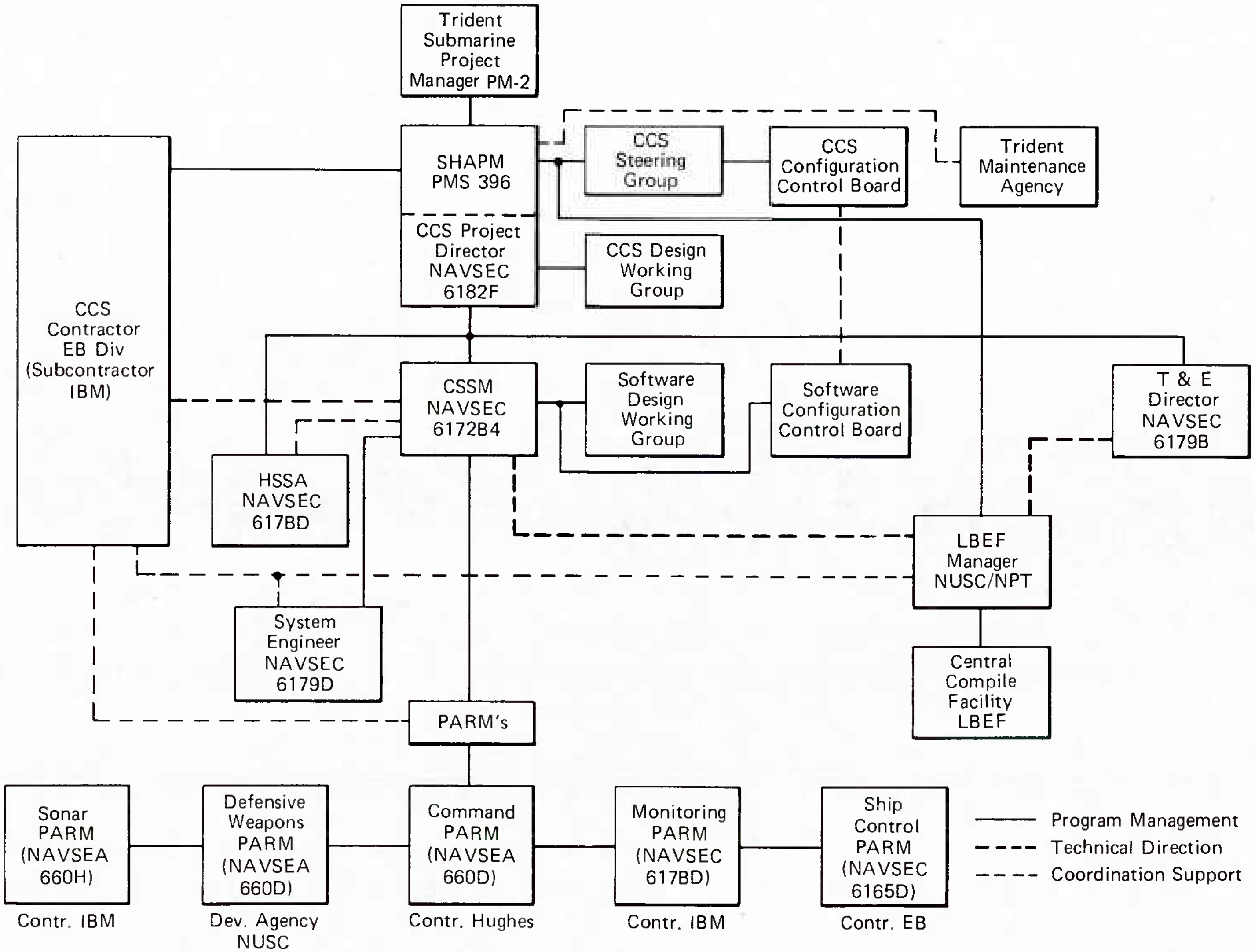


Fig. 2-5 CCS Software Agencies and Reporting Relationships

1. Direction of computer program development, and

2. Direction of computer program integration and verification.

2.6.2.2 CCS Contractor

As indicated in Section 2.4.1, Electric Boat (as the major contractor) has subcontracted with IBM to assist in carrying out all relevant software activities. These include project coordination, technical support and technical management, design, engineering, and development of designated software, integration test and evaluation, and demonstration of the integrated CCS software. Thus, CCS contractor responsibilities lie in four primary areas:

1. Project Planning, Coordination, and Support;

2. Design and Engineering;

3. Software Specification and Development; and

4. Integration and Testing.

2.6.2.3 Hardware and Support Software Agent (HSSA)

This agent was described in subsection 2.4.3.1.

2.6.2.4 CCS Test and Evaluation (T&E) Director

The CCS T&E Director NAVSEC 6179B assigned by PMS-396 is directing the activities of the LBEF organizations and providing the necessary coordination with the CSSM.

The T&E Director's responsibilities will also include:

1. Resolving major technical problems and providing overall guidance to the CCS T&E effort;

2. Monitoring specification of verification software; and

3. Directing the Test and Evaluation of the integrated Hardware/Software System.

2.6.2.5 Land Base Evaluation Facility (LBEF) Manager

The LBEF Manager's prime responsibilities are:

1. Preparing LBEF verification system (simulation/stimulation) test hardware procurement specifications;

2. Constructing the facility;

3. Installing and testing the LBEF test hardware support system;

4. Installing, verifying, and maintaining CCS cabling and equipment at the LBEF; and

5. Providing facility support and personnel required for CCS hardware and test equipment maintenance, facility/program security services, equipment calibration, quality control, logistics, and configuration status accounting.

2.6.2.6 Certification Agent

NUSC/NPT is the designated agent of PMS-396 for certification of the CCS at the LBEF. The Certification Agent is responsible for:

1. Certification of the LBEF Simulation/Stimulation Support System (hardware and software);

2. Certification of the CCS; and

3. Documentation for certification of the CCS.

2.6.2.7 Participating Managers (PARM's)

Each of the CCS Participating Managers (PARM's) and their respective contractors will be responsible for generation and qualification of their programs and for providing support to the CSSM in the planning, integration, and acceptance activities. The three major areas of PARM responsibilities are:

1. Planning and design support to CSSM;

2. Computer program generation and qualification; and

3. Support for software integration and verification, CCS test and evaluation, and shipboard installation and acceptance.

PARM's are also referred to as Principal Development Agencies in some Trident documentation.

2.6.2.8 Software Design Working Group (SDWG)

The Trident Submarine CCS SDWG acts in an advisory capacity to the CSSM and assists, as directed, in matters related to design, development, integration, and testing of the CCS software. The purpose of this group is to ensure that CCS software problem areas are identified early and that effective action is taken by appropriate participating agencies. The SDWG is chaired by the CSSM and is composed of officially designated representatives from PMS-396, the CCS Contractors, HSSA, Test and Evaluation Director, LBEF and the PARM's and their development representatives.

2.6.2.9 Software Configuration Control Board (SCCB)

The SCCB is a decision-making body composed of designated representatives from the software agencies in Fig. 2-5. Computer programs are retained under PARM Change Control until subsystem qualification. The SCCB will be responsible for directing configuration control over software, extending from qualification through software acceptance aboard ship. After OPFCO certification, the Maintenance Agency will take over the configuration control functions for the remainder of the system's life.

Specific SCCB responsibilities include:

1. Coordination of change proposals between all users;

2. Coordination of software changes with Trident Configuration Control Board (CCB) actions to ensure software and software/hardware interface compatibility;

3. Recommendations to the Trident CCB which involve hardware/software and system software change tradeoffs (i.e., those instances where the same result can be obtained in several ways); and

4. Initiation of implementation of approved software change proposals.

2.6.2.10 Computer Program Maintenance Agency

Maintenance Agency definition is currently in the planning stage. A new, dedicated Trident maintenance agency will be established at Bangor, Washington.

Prime responsibilities of the Maintenance Agency will be:

1. Responding to shipboard and shore-based program problem reports;

2. Reviewing and commenting upon Engineering Change Proposals;

3. Testing program modifications for system impacts; and

4. Delivering of all program patches and updates to ships

2.6.3 Management and Specification Documents

Figure 2-4 lists the relationship between primary CCS management and specification documents. Provision was made for a progressively more detailed formal documentation chain in both the specification and the management areas.

A comprehensive list of government and nongovernment documents pertinent to the production of the major documents indicated in Fig. 2-4 is given in Table 2-6. (One subset of this total list was shown in Table 2-4.) These documents were specified as applicable to Trident Software Development in the CCS Software Management Plan.

2.6.4 Management Techniques

Various aspects of the management of Trident Software development, beginning with the Software Management Plan Document, merit highlighting. (Some parts of this Management Techniques section were excerpted in Sections 2.4 and 2.5; full listings are given here for convenience.)

2.6.4.1 Software Management Plan

A formal CCS Software Management Plan was prepared which described:

1. Responsibilities for performance of tasks defined for CCS computer program development in relation to the overall management structure for the Trident Submarine;

2. The development approach, particularly in the activities associated with computer program generation and integration;

3. The CCS in the context of the CCS and ship subsystem interfaces with the CCC, the CCC equipment configuration, and the operational software modules of the CCS;

TABLE 2-6

CCS SUPPORTING DOCUMENTS

Document	Title
MIL-STD-12	Abbreviations for Use on Drawings
MIL-STD-480	Configuration Control Engineering Changes, Deviations and Waivers
MIL-STD-483	Configuration Management Practices for Systems, Equipments, Munitions and Computer Programs
Weapons Specification WS-8506, Revision 1	Requirements for Digital Computer Program Documentation
MAT 09Y: EWC Ser 23	TADSTAND 1 - Standard Shipboard Tactical Digital Computer and Program Language
MAT 09Y: RSF Ser 25	TADSTAND 2 - Standard Specification for Digital Computer Documentation
MAT 09Y: CFH Ser 33	TADSTAND 3 - Standard Requirements for Inter-Digital Data Processor Interface Documentation
MAT 09Y: CFH Ser 113	TADSTAND 4 - Standard Definition of Tactical Digital Systems
MAT 09Y: CFH Ser 134	TADSTAND 5 - Standard Reserve Capacity Requirements for Digital Combat System Processors
MAT 09Y: EWC Ser 148	TADSTAND 6 - Combat System Designs Employing Multiple AN/UYK-7 Processor
Trident Submarine Command and Control System Management Plan, 1 July 1972	
NAVSHIPS 0905-497-1010	System Operational Specification for Trident Submarine CCS, 15 July 1973
Command and Control System Test and Evaluation Management Plan, March 1973	
NAVSHIPS 0967-029-5122	Program Specification for Common Program MOD X2, 15 January 1973
NAVSHIPS 0967-029-5152	Computer Program Performance Specification, Common Program MOD X2 Standard Executive (Preliminary) 1 August 1973

TABLE 2-6 (Cont'd)

Document	Title
NAVSHIPS 0967-029-5154	Computer Program Performance Specification, Common Program MOD X2 Common Peripheral Module (Preliminary) 1 August 1973
NAVSHIPS 0967-029-5155	Computer Program Performance Specification, Common Program MOD X2 Dynamic Module Replacement Module (Preliminary) 1 August 1973
NAVSHIPS 0967-028-0600	Users Reference Manual for Compiler Monitor System (CMS-2) for use with AN/UYK-7 Computer, November 1971
NAVSHIPS 0967-051-6291	Specification for Digital Computer AN/UYK-7
NAVSHIPS 0967-024-5450	Operating Procedures for Computer Set AN/UYK-7 Diagnostic Programs
NAVSHIPS 0900-074-4010	Trident Command and Control System Specification (latest date of issue)
NAVSHIPS 0900-076-4010	Interface Design Specifications (IDS) Functional Interface Definitions, 14 December 1973
NAVSHIPS 0900-074-3010	Trident Submarine CCS Configuration Management Plan (latest date of issue)
NAVSHIPS 0900-075-7040	Command and Control System Trident Submarine Design Data Document, Volume IV, Part 1, Software Design and Interface Description, 31 August 1973
NAVSHIPS 0900-076-0010	Trident Submarine Hardware Scheduling Plan for Central Computer Complex (latest date of issue)
NAVSHIPS 0900-076-5010	Trident CCS Software Standards and Conventions (latest date of issue)

4. Computer programs to be required for support of software development, shipboard operation in the Central Computer Complex, integration test and evaluation of CCS system hardware and software, and the ongoing mission data analysis and maintenance activities ashore;

5. A software development plan segmented into five overlapping phases including the schedules established for development activities and deliverable items;

6. Provisioning of computer program development and checkout facilities; and

7. Software development standards and constraints which are imposed on all computer program developers to ensure adequate means of achieving management visibility and controls required for integration, including provision for standard program documentation and development status reports.

2.6.4.2 Software Development Plan

Five phases were defined for the CCS software acquisition/development process. The defined phases cover the complete process, from initial software definition through training and maintenance. Following paragraphs summarize each phase.

Phase I - Planning

Key elements of this phase were:

1. CCS Software Definition;

2. Central Computer Complex Design;

3. Standards, Conventions, and Constraints;

4. Integration Test and Evaluation Plans;

5. Management, Support, and Maintenance Plans; and

6. Subsystem Management and Performance Plans.

Phase II - Computer Program Generation

Key elements defined for this phase were:

1. Program Analysis;

2. Program Design;

3. Top-Down Structured Program Generation;

4. Delivery Requirements for Integration — Intermediate deliveries - Level 1, Level 2, Partial Programs, and Complete Programs;

5. Computer Program Documentation; and

6. Program Qualification.

Phase III - Computer Program Integration Test & Evaluation

The qualified subsystem programs will be the primary software elements to be integrated at the LBEF. During this phase, the verified software will be subject next to system-level (hardware/software) tests and then to LBEF certification performed by NUSC/Newport (as agent of PMS-396).

Phase IV - Shipboard Installation and Acceptance

The shipboard installation of the CCS Computer Program will commence following successful completion of integration and verification tests at LBEF. Furthermore, continued shipboard software integration support will be provided by the CCS contractors during this phase.

Phase V - Computer Program Maintenance

The culmination of the at-sea test activity will be the acceptance of the integrated program by the maintenance agent. Maintenance of the integrated program will involve the four prime responsibilities of the maintenance agency as described in subsection 2.6.2.10.

2.6.4.3 Phased, Milestoned Deliveries

As indicated in Section 2.5.4, phased deliveries permit early problem identification among subsystem programs and between the subsystem programs and the operating system. The results of the integration of phased deliveries are promulgated via CCS Software Status Reports. Thus, these phased, milestoned deliveries together with the associated Software Status Reports form a major management tool for progress-monitoring.

2.6.4.4 Quality Assurance

Organizations such as the Software Design Working Group (SDWG) (subsection 2.6.2.8) and the Software Configuration Control Board (SCCB) (subsection 2.6.2.9) contribute to improved configuration control and hence improved quality assurance potential for Trident software. Similarly, techniques and procedures such as Top-Down Design/Structured Programming (subsection 2.4.3.1), Integration Testing (section 2.5.4), and Certification Testing (section 2.5.5) also provide enhanced quality assurance.

2.6.4.5 Documentation Requirements

An early (and major) management decision is to settle on the required documentation. In the case of Trident, basic standards applied to software documentation are the WS-8506 for Digital Computer Program documentation and TADSTAND No. 3, Standard Requirements for Inter-Digital Data Processor Interface Documentation.

Supplemental documents specified for the support of Trident development were:

1. Computer Program Performance Specification,
2. Computer Program Design Specification,
3. Function Operational Design,
4. Computer Subprogram Design Document,
5. Computer Program Test Plan,
6. Computer Program Test Procedures,
7. Position Operating Procedures, and
8. Computer Program Operations Manual.

2.7 OPERATIONAL SOFTWARE MAINTENANCE

2.7.1 Computer Program Maintenance Agency

As discussed earlier (subsection 2.6.2.10), a Maintenance Agency will be established at Bangor, Washington to (a) respond to program problem reports, (b) review and comment upon Engineering Change Proposals, (c) test program modifications, and (d) deliver all program patches and updates.

2.7.2 Maintenance Support Software

This class of software configuration items consists of problem modules to provide automated assistance to preventive maintenance, fault detection, and fault isolation functions for CCC equipments in the form of diagnostic programs.

2.7.3 Training Support Software

Appropriate program modules for shorebased and/or on-board training of personnel in CCS operation and maintenance is provided based on a "Training Software Plan," as specified by the Software Management Plan.

2.8 HIGHLIGHTS

Trident treats operational software and Test and Integration software as configuration items. The compiler, CMS-2Y, was furnished by the Government and not called out as a deliverable. (MP3)

The Trident Software Management Plan defined five development phases with specific deliverables. The development phases were: planning, program generation, integration test and evaluation, shipboard installation and acceptance, and program maintenance. Deliverables included programs, documentation, plans, and test facilities. (AP1)

Subsequent to the Proposed Technical Approach (PTA) study, early software management planning defined schedules, organization, standards and conventions, implementation methods, operating philosophy, integration strategy, resource control/ allocation, and the Land-Based Evaluation Facility. (AP1, AP2)

Specified design and implementation techniques such as top-down design and structured programming allowed defined and phased delivery of software components, thus permitting early interface/integration testing. (AP1, SE3, IP2)

Prior to DSARC II, Trident conducted an in-depth, detailed Proposed Technical Approach (PTA) study that examined both centralized and decentralized systems. The PTA defined four viable alternatives. The Navy Ship Acquisition Manager (with PM-2 approval) then selected a centralized computer system concept. (SE1)

Trident was the first Navy system to specify top-down design. The use of top-down design was addressed in the Command and Control System (CCS) Software Management Plan. The requirement included not only the application of the procedure to the overall system, but also independently to the major modules of the subsystem. (SE3)

The Trident Software Management Plan specified a disciplined set of programming practices, including:

1. Use of CMS-2Y language;

2. Use of structured programming by all developers;

3. Production of a software standards and conventions manual featuring design conventions, naming standards and conventions, coding techniques, etc.; and

4. Use of WS 8506 as a documentation standard .

(IP2)

Trident planned for and implemented a Land-Based Evaluation Facility during the conceptual phase of development. It has the specific purpose of software development, test, integration, and certification. The facility includes tactical equipment as well as computer systems facilities. The facility supports total systems integration and verification of each tactical equipment suite prior to shipboard installation.
(IP3)

3. PERSHING WEAPON SYSTEM

3.1 GENERAL SYSTEM DESCRIPTION

The Pershing Ia is a mobile, nuclear ballistic missile system. It is composed of all firing battery components required to conduct launch operations as well as equipment necessary for rear area support and maintenance functions. It is presently deployed in Europe.

The Pershing Weapon System has two basic missions:

1. QRA: The Quick Reaction Alert (QRA) role is one of a high state of readiness with predetermined assignments to high priority targets.

2. General Support: After the QRA mission is completed, Pershing reverts to the Field Army Commander's control for general support of the field Army.

3.1.1 Pershing Subsystems

Major equipment items required at a launch site are the erector-launcher, power station, programmer test station (PTS), missile, azimuth laying equipment, and battery control central. This equipment provides the basis for independent operations at a forward launch area and performs the final sequencing and launch of the missile.

The Pershing system contains two major computer systems, the PTS computer and the Guidance and Control Computer (G&CC) on board each missile.

The PTS functions as the mobile fire control center. PTS equipment controls prelaunch sequence, computes missile presets and firing azimuth, and provides communiations, power distribution and control. The PTS also provides basic test and malfunction analysis, including fault isolation and self-test capabilities. Each PTS can support three missiles.

The system components test station (SCTS) is used in performing rear area maintenance. As a housed mobile center, the SCTS uses a computer and tape programs for testing missile sections, together with assemblies, cards, relays, and modules from the guided missile and associated ground support equipment. Diagnostic tape programs are also provided for verification and troubleshooting of the major SCTS assemblies. The SCTS contains a dismounted PTS and has facilities whereby one missile guidance section can be tested and another repaired simultaneously under controlled temperature conditions.

3.1.2 Acquisition History

The Pershing Weapon System was developed and deployed during the 1958-1964 period. The Pershing Project Manager's Office (PPMO) has controlled development and generated the System Specification. MIL-STD's 480 and 490 have been used to control configuration management procedures and documentation.

3.1.3 System Block Diagrams

Figure 3-1 shows the schematic of the Pershing Ia Weapon System.

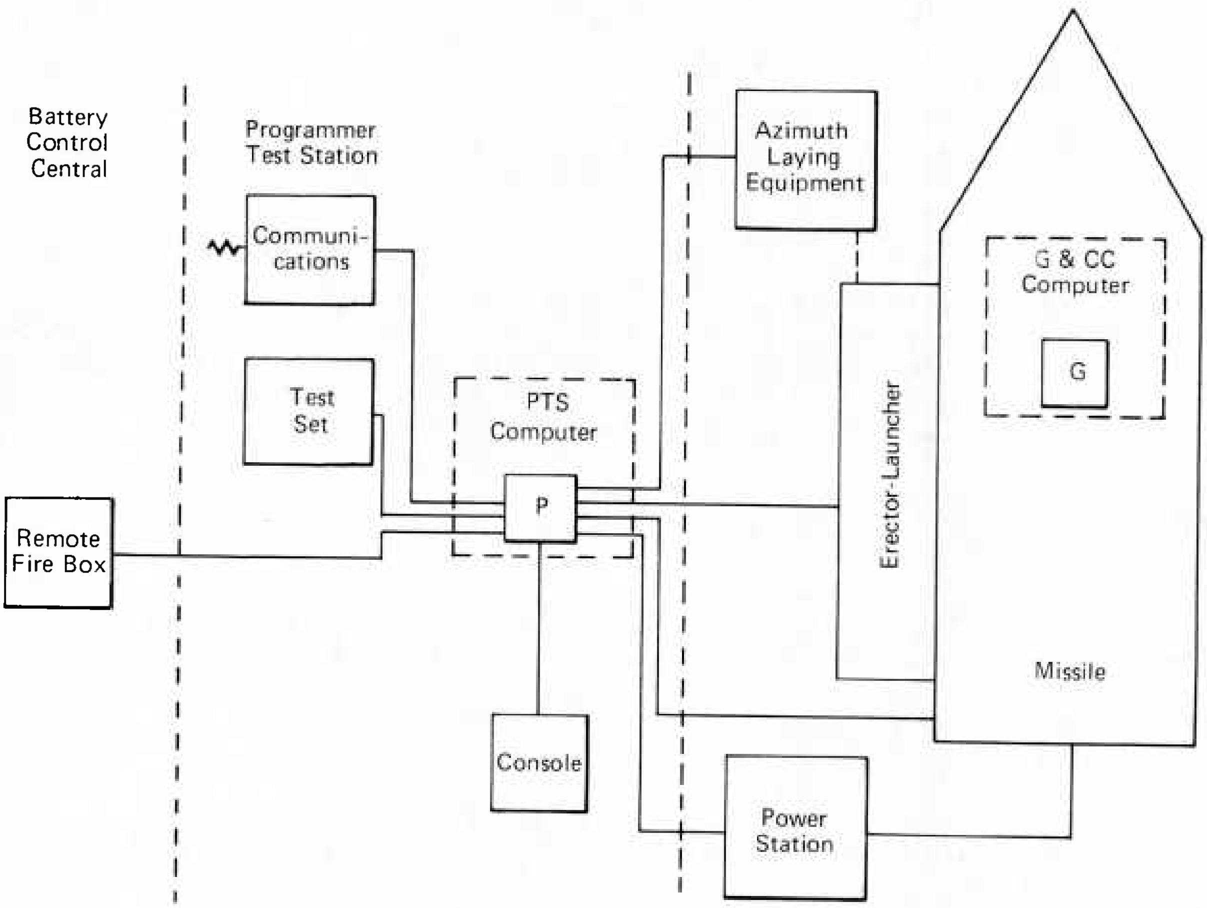


Fig. 3-1 Pershing Weapon System

A diagram of the interface and signal flow for the PTS is shown in Fig. 3-2. The overall block diagram for the SCTS is shown in Fig. 3-3.

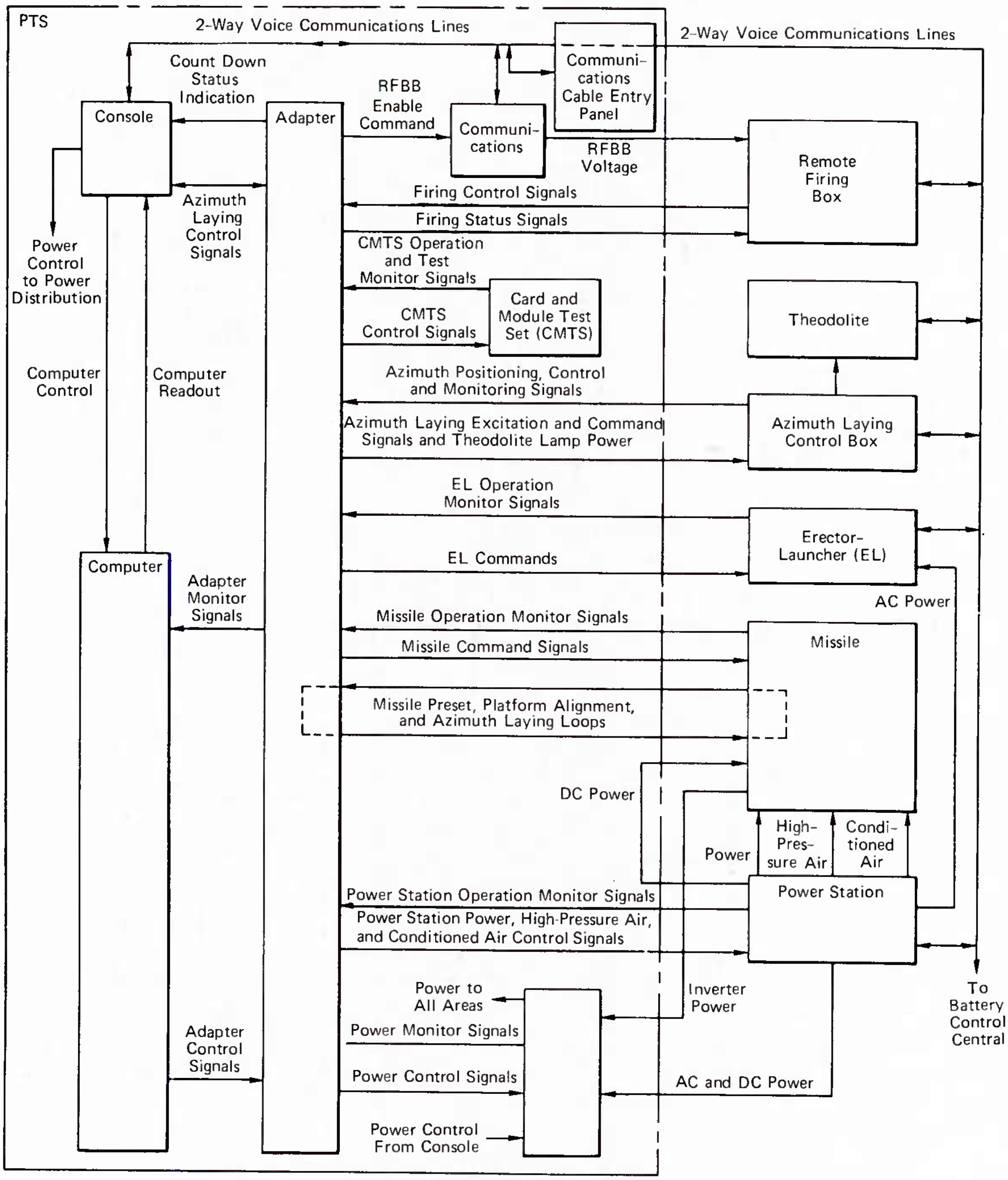


Fig. 3-2 PTS Block Diagram

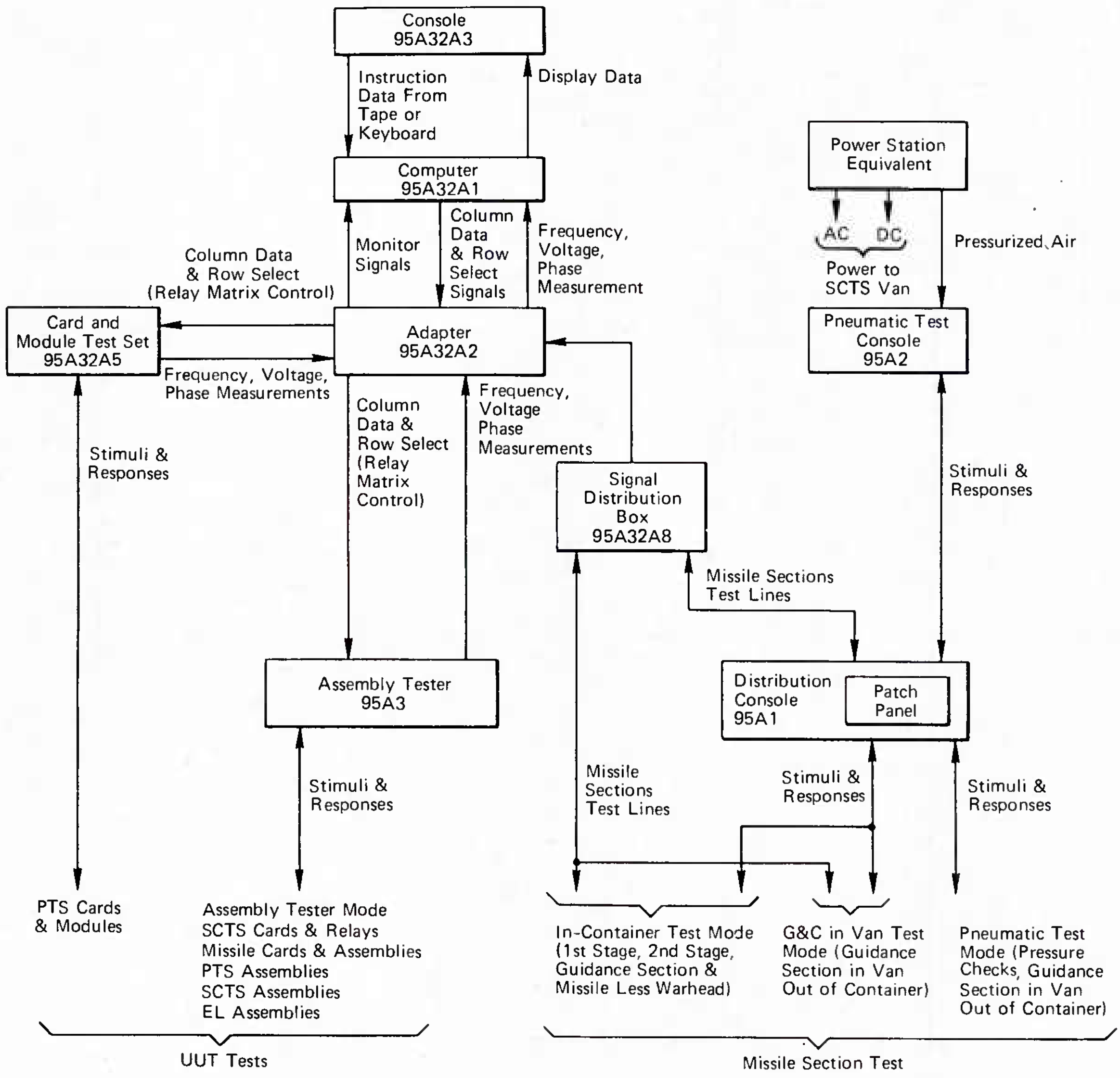


Fig. 3-3 SCTS Overall Block Diagram

3.2 COMPUTER SYSTEM ARCHITECTURE

3.2.1 General Description

As indicated, the Pershing system contains two major computer systems. The PTS functions as the mobile fire control center and as such is supported by a computer system consisting of a general purpose computer, maintenance panel, countdown panel, tape reader and teleprinter. Each Programmer Test Station can support three missiles, each of which contains a Guidance and Control Computer (G&CC). The G&CC computer interfaces with the ST 120 stabilized platform, main distributor, telemetry, ground support equipment, and the hydraulic actuators. The G&CC and PTS computers communicate with a bit-by-bit data link. A summary of the two computers (keyed to Fig. 3-1) is given in Table 3-1.

TABLE 3-1

PERSHING COMPUTER SUMMARY

Unit	Type	Function	Processors	Memory
P	Burroughs D84 (24 bit, 4 μs)	Target data entry, firing sequence control, missile preset, status monitoring	1	16k
G	Bendix BDX 820 (16 bit, 2 μs)	Missile in-flight guidance and control	1	4k

3.2.2 PTS Computer

Figure 3-4 is a block diagram of the ground computer. The computer itself is a 24-bit, 4 μsec, 16k unit manufactured by Burroughs for the Pershing system. The primary functions performed are manual entry of target information, initiating, controlling and monitoring the firing sequence, and the generation of command signals for the adapter. The adapter converts digital computer inputs and outputs to analog signals for presetting the missile guidance section, monitoring and controlling the ground support equipment and checking the status of missile sections. The primary interaction with the operator is through the special purpose countdown and maintenance panels. The teleprinter records all information displayed on the countdown panel. The tape reader is used to load computer programs into both the PTS and G&CC computers.

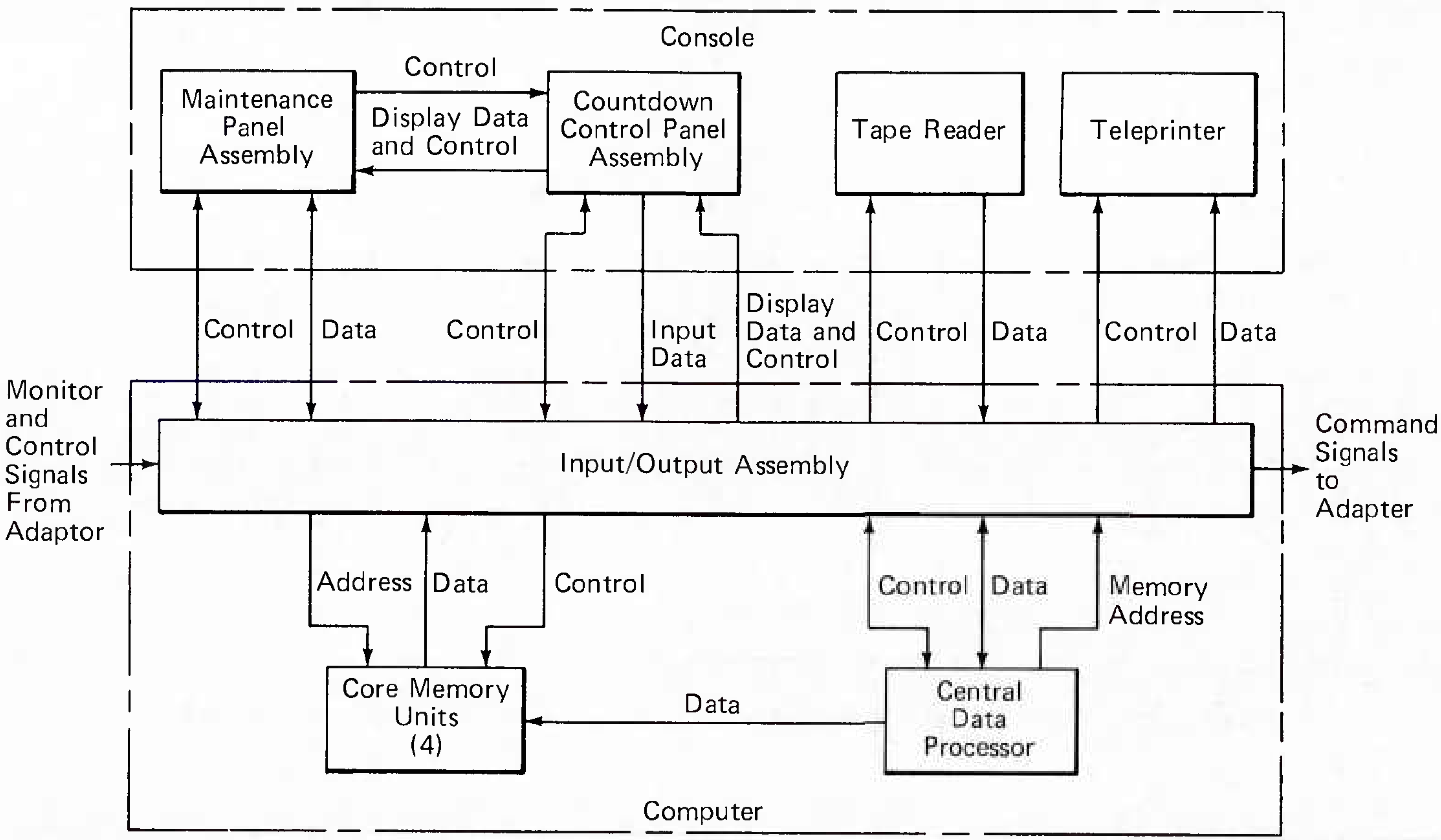


Fig. 3-4 PTS Computer Block Diagram

3.2.3 Guidance and Control Computer

Unlike the PTS computer which has been a part of the original Pershing Ia system, the Guidance and Control Computer in the missile was added in the latest modification to the missile system. Thus, this computer is somewhat more technologically advanced. The processor is a BDX 820 manufactured by Bendix with a basic cycle time of 2 μsec. The memory size is 4k of 16-bit core plus a 64-word read-only memory. Since this system was all analog in earlier versions, the Input/Output Processor of the computer has been especially tailored to an analog environment.

The airborne processor controls the missile during flight, accepts the presets from the PTS and interfaces with the guidance package sensors and instrumentation.

3.3 COMPUTER PROGRAM ARCHITECTURE

The tactical computer software employed in the Pershing system may be divided into two major categories:

1. Ground computer software (primarily PTS), and
2. Guidance and control computer (G&CC) software.

Of these two, the ground computer software is by far the most extensive, including a myriad of test programs in addition to the tactical countdown programs. The G&CC computer software consists of one major computer program containing both a ground phase and an airborne phase. During a missile countdown, the ground phase of the G&CC program acts as a slave to the PTS countdown program, assisting it in performing ground checkout of the missile. The G&CC program also accepts the missile guidance and warhead fuzing presets, necessary for flight, which are computed by the PTS countdown program.

3.3.1 Ground Computer Software Architecture

The Pershing ground computer software consists of 38 major program tapes, 14 of which may be classified as "forward area" tapes (i.e., available to troops in the field) and the remaining 24 "rear area" tapes (available to support personnel only).

All but three of the forward area programs are diagnostic, testing, or training programs. These programs may be executed by either the PTS computer or the SCTS computer. The remaining three programs are the actual tactical countdown programs written specifically for a given system configuration and modification level. Only one of the three countdown programs is used, depending on the configuration at the particular site.

An individual countdown program has three major options — standard, confidence, and quick. The confidence and standard countdowns are comprehensive, testing the great majority of system functions. Of these two, only the standard countdown may be used to launch a missile. The confidence countdown is used at regular intervals to maintain confidence in junctions not tested by the quick countdown, which is the option normally used to launch a QRA missile. A quick countdown may be performed only after a confidence countdown has been successfully completed at some previous time.

There are 34 basic subfunctions in the countdown program. Of these, 22 must be serially executed to fire a missile in the quick mode. In the standard mode, seven additional subroutines are executed while only one is eliminated. In addition, in the standard mode, some subroutines are required to perform additional tests. A sequential list of the 34 subfunctions used in the 11053062-3 countdown program along with their basic purpose is given in Table 3-2. No documentation exists to relate each of these subfunctions easily to storage requirements in the PTS computer. However, the entire program is stored in the four 4096-word memory units in the present PTS. An additional memory unit is being added to accommodate the Pershing Block VII modification to the system.

3.3.2 G&CC Software Architecture

The G&CC computer program may be divided into a ground phase and an airborne phase. During the ground phase, the PTS countdown program diverts the G&CC program from the self test mode into individual ground tests, such as simulated flight, accelerometer tests, and control tests.

In the airborne phase, the G&CC program performs six basic functions:

1. Guidance: Accepts missile velocity signals in the form of accelerometer synchro voltages, and outputs yaw and pitch steering signals to the control function and a cutoff signal to the good guidance function;

2. Stability: Accepts missile attitude signals from the inertial reference, demodulates and filters them and converts them to digital words, which are then processed using a time-dependent transfer function;

TABLE 3-2

11053062-3 COUNTDOWN SUBFUNCTIONS

Subfunction	Purpose
PROGRAM ENTRY	Enters the countdown program tape
COMPUTER SELF TEST	Tests PTS computer, verifies program loading
MANUAL DATA ENTRY	Accepts operator inputs, computes presets
ADAPTER POWER ON	Verifies adapter modules, prints trajectory data
ADAPTER SELF TEST	Tests adapter circuits
INITIAL MONITOR	Verifies missile status, enables monitors
MISSILE POWER ON	Turns on missile power, verifies bus activation
INITIATE G&CC	Verifies initial status of G&CC
RANGE SELECT	Verifies missile range and confidence count status
WARHEAD PRESET AND VERIFICATION	Presets and verifies warhead synchros, zeroes pitch program
CONTROLS TEST	Sends test commands to missile, monitors feedback
PLATFORM ENABLE	Uncages ST-120 inertial platform
VANE LOCK	Locks the second-stage vanes
CROSS RANGE OFFSET TEST	Verifies cross range accelerometer and roll microsyn
SIMULATED FLIGHT	Tests G&CC for discrete times and cutoff solution
SET ERB	Verifies platform temperature, sets ERB pots
AUTO SLEW	Torques ST-120 to perdetermined position
AZIMUTH LAY COARSE	Allows theodolite operator to torque ST-120

TABLE 3-2 (Cont'd)

Subfunction	Purpose
AZIMUTH LAY FINE	Allows operator to perform final positioning of ST-120
OPTICAL MONITOR	Allows operator to monitor platform drift
ACCELEROMETER MONITOR	Checks precession rates of three missile accelerometers
PRESET FLIGHT	Presets G&CC to computed flight values
BURST SELECT	Verifies and sets warhead burst option
ERECTION	Verifies EL status, erects missile, activates missile pumps
AZIMUTH ROTATION	Allows ST-120 to be rotated in vertical lay mode
PRE-REMOTE MONITOR	Checks status of system
REMOTE	Allows time for personnel evacuation, makes final tests
LAUNCH SEQUENCE INITIATION	Starts launch sequence
FIRE SEQUENCE	Performs final preparation, power transfer, sends ignition commands
MONITOR SCAN	Monitors system safety circuits once per second
MISSILE POWER OFF	Secures missile power for shutdown
EMERGENCY INTERRUPT	Determines cause of emergency interrupt, powers down
FAULT ISOLATION	Isolates malfunction to component level without intervention
MALFUNCTION	Stores and displays malfunction and test number

3. Control: Accepts signals from the guidance and stability functions and processes them to form command signals for the vane control hydraulic actuator packages;

4. Time discrete generator: Issues prescribed signals in real time for use by time-dependent functions;

5. Good guidance: Monitors velocity errors, displacement, voltage, and pressure to generate good guidance signal; and

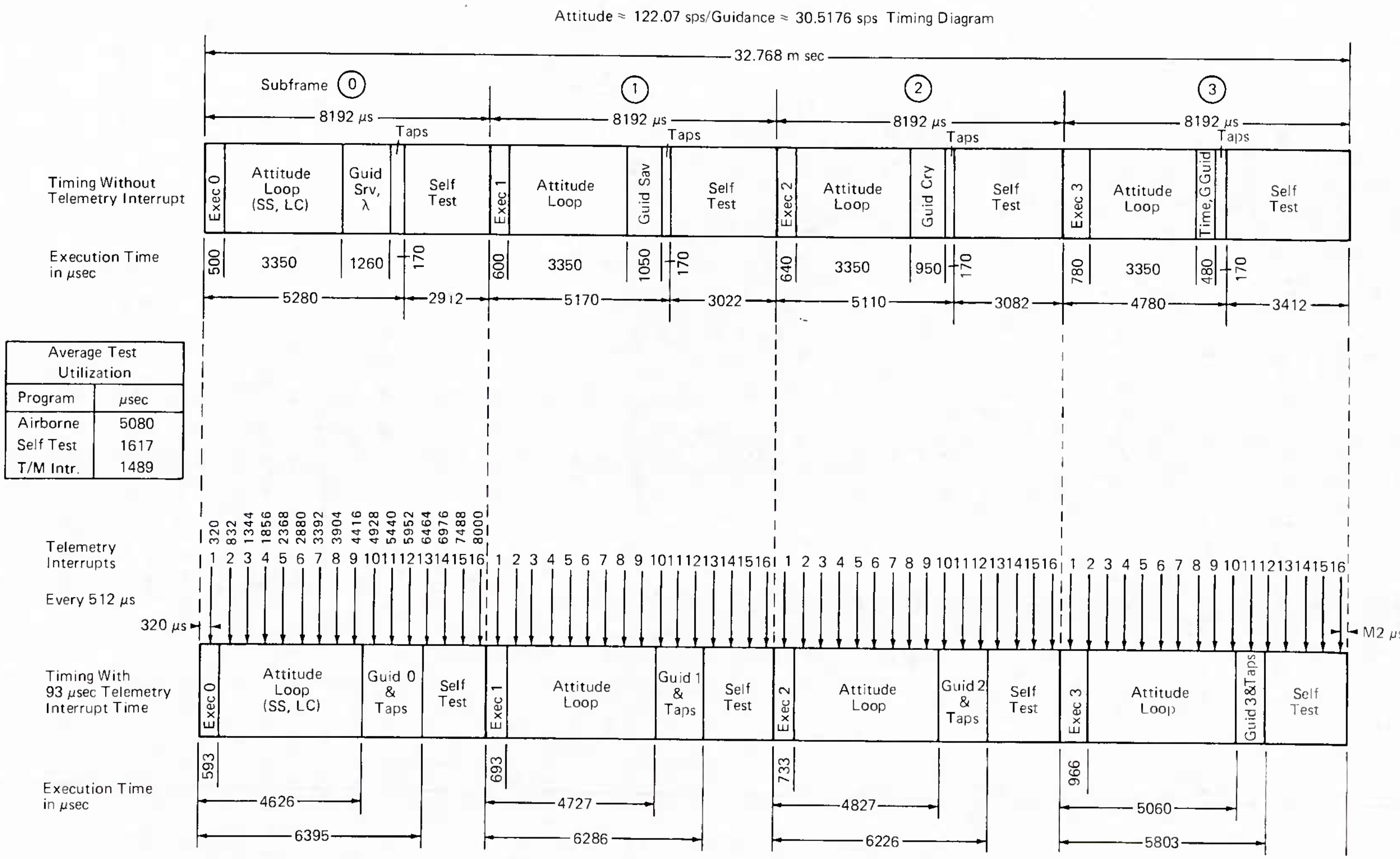
6. Telemetry: Outputs prescribed missile functions to telemetry.

During the airborne phase, the G&CC program operates as a real-time system reliant on interrupts to control processing. A real-time interrupt is generated every 8.192 ms which directs the program to perform one of four sequences of subprograms, so that all functions are performed at least once every 32.768 ms. When a sequence has been performed, the program returns to self test, remaining in that mode until the next interrupt. Telemetry interrupts are generated every 512 μs. Figure 3-5 is an illustration of the timing requirements for the present version of the G&CC airborne computer program. This figure indicates that all calculations may be performed in 62% of the available time, while 18% is required for telemetry processing, and 20% remains allocated to self test.

Table 3-3 gives a summary of the storage requirements for both the ground and airborne phases of the G&CC computer program. The -7 version is the version presently deployed and the -13 represents the version to be fielded with the Block VII modification to the missile system. It should be noted that all versions remain well within the 4096 word capacity of the BDX 820 computer.

3.4 SOFTWARE DEFINITION, DESIGN, AND IMPLEMENTATION

The Pershing weapon system was developed and deployed during the 1958-64 period and therefore did not use prescribed government standards for software. Rather, software was treated like hardware during the engineering phase and documented accordingly. At the present time, the PPMO uses the MIL-STD-490 Configuration Management Program techniques for the two most critical software programs used in the Pershing Weapon System. The two programs elected are the airborne flight and the fire control system countdown programs. The techniques used for software definition for these two programs will be explained in detail in this section. The other 147 software programs used in the Pershing Weapon



Average Test Utilization	
Program	μsec
Airborne	5080
Self Test	1617
T/M Intr.	1489

Fig. 3-5 G&CC Airborne Program Timing

TABLE 3-3

G&CC COMPUTER PROGRAM STORAGE REQUIREMENTS

Routines	PIa G&CC Program Usage		
	-5	-7	-13
I. GROUND PROGRAM			
1 NDRO	128	128	128
2 DATA, LINKS, INTRRPT	174	199	199
3 PTS COMMUNICATION	149	149	149
4 VANE DITHER	57	57	57
5 SELF TESTS	848	848	848
6 ST MON/SIM FLT	149	137	137
7 ACCEL TESTS	42	51	51
8 MISC GND TESTS	19	23	24
9 CONTROLS TEST	78	71	71
10 A/D - D/A RTNS	30	30	30
II. AIRBORNE PROGRAM			
1 EXECUTIVE	174	194	194
2 TELEMETRY FORMAT	174	159	159
3 SLANT RANGE	67	67	67
4 SLANT ALT & CROSS RANGE	85	84	122
5 TIME FUNCTION GEN	274	292	330
6 GOOD GUIDANCE	70	70	64
7 AUTOPILOT	282	282	282
8 SYNCHRO CONVERSION	122	109	109
9 ARITHMETIC SUBR	64	78	78
10 LIFTOFF INTRR	154	155	147
11 T/M - TAPS OUT	27	27	17
12 TAPS TEST	102	124	0
13 DATA & LINKS	194	195	187
III. TEST STAND LOADER	49	49	49
TOTAL	3514	3578	3499

System did not follow the MIL-STD-490 requirements exactly because of the cost involved, but they too follow quality assurance and documentation procedures which appear to be adequate.

3.4.1 Software Definition

It is during this time that the requirements specifications are written to define what the system, hardware, program must do. For Pershing this is done as follows. (See Fig. 3-6 Pershing Specification Tree.)

1. The MIL-STD-490 Type A System Specification for Pershing (MIS-11550) is written by PPMO.

2. The MIL-STD-490 Type B Development Specification for the components of the system (including the two controlled software items, MIS-11560 and MIS-11565 (Mandatory Item Specification)) is prepared by the contractor Martin Marietta Aerospace (MMA) and submitted to the PPMO for approval or change.

The two quoted documents are written exactly as required for the MIL-STD-490 Type B5 Part I "Computer Program Development Specification". Table 3-4 is a documentation outline of MIS-11565 for the PTS computer software program.

3.4.2 Software Design

Pershing specifications use the two-part option explained in detail in Section 3.1.4 of MIL-STD-490. During the design phase, requirements are assigned to specific software modules, and functional block diagrams are developed to define how the job is to be done.

1. PTS Software Design Phase: For the PTS software, MIS-11541 was prepared by the contractor and submitted for approval to PPMO.

2. Airborne Software Design Phase: The airborne software design was done using the two-part specification (Type C5). A unique technique for Pershing was applied at this stage. Figures 3-7 and 3-8 are examples of software design being adapted to engineering drawings with all the controls normally associated with hardware design drawings. The intent was to flow the program such that nonprogrammers could easily follow the software logic. Figure 3-9 is the next lower level and is intended for use by the actual programmers.

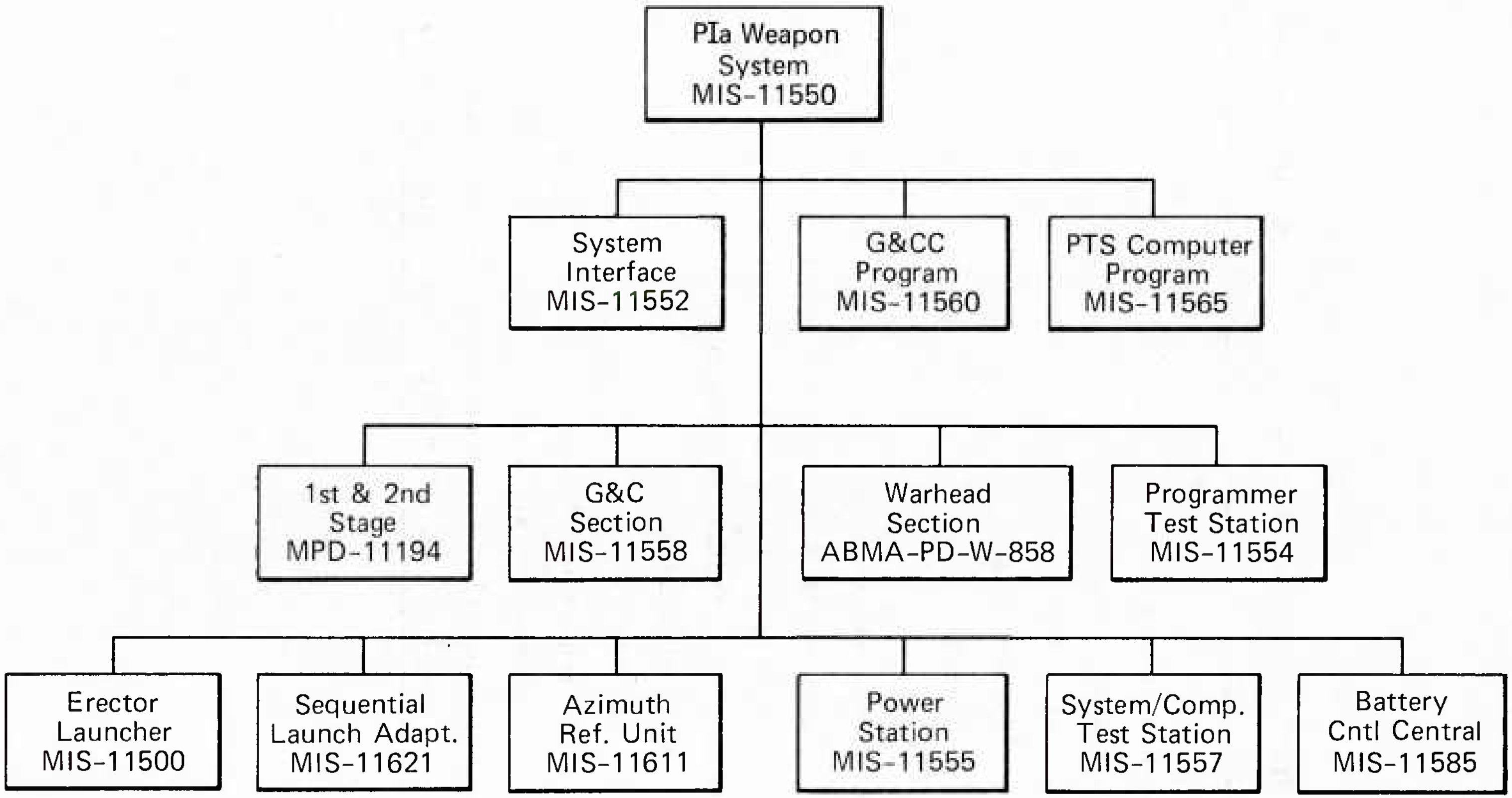


Fig. 3-6 Pershing Specification Tree

TABLE 3-4

DOCUMENTATION OUTLINE OF MIS-11565 FOR THE PTS COMPUTER PROGRAM

- Scope
- Applicable Documents
- Requirements
  - Program Interfaces
    - Equipment
    - Software
    - Human
  - Functional Requirement
    - Interfacing Equipment & Program
    - CDN Sequences & Options
  - Detailed Functional Requirements
    - CDN Subfunctions
    - CDN Organization
    - CDN Sequence Options
- Quality Assurance Provision
- Appendix I - Detailed CDN Subfunction Requirements
- Appendix II - Detailed CDN Subfunction Sequence Requirements

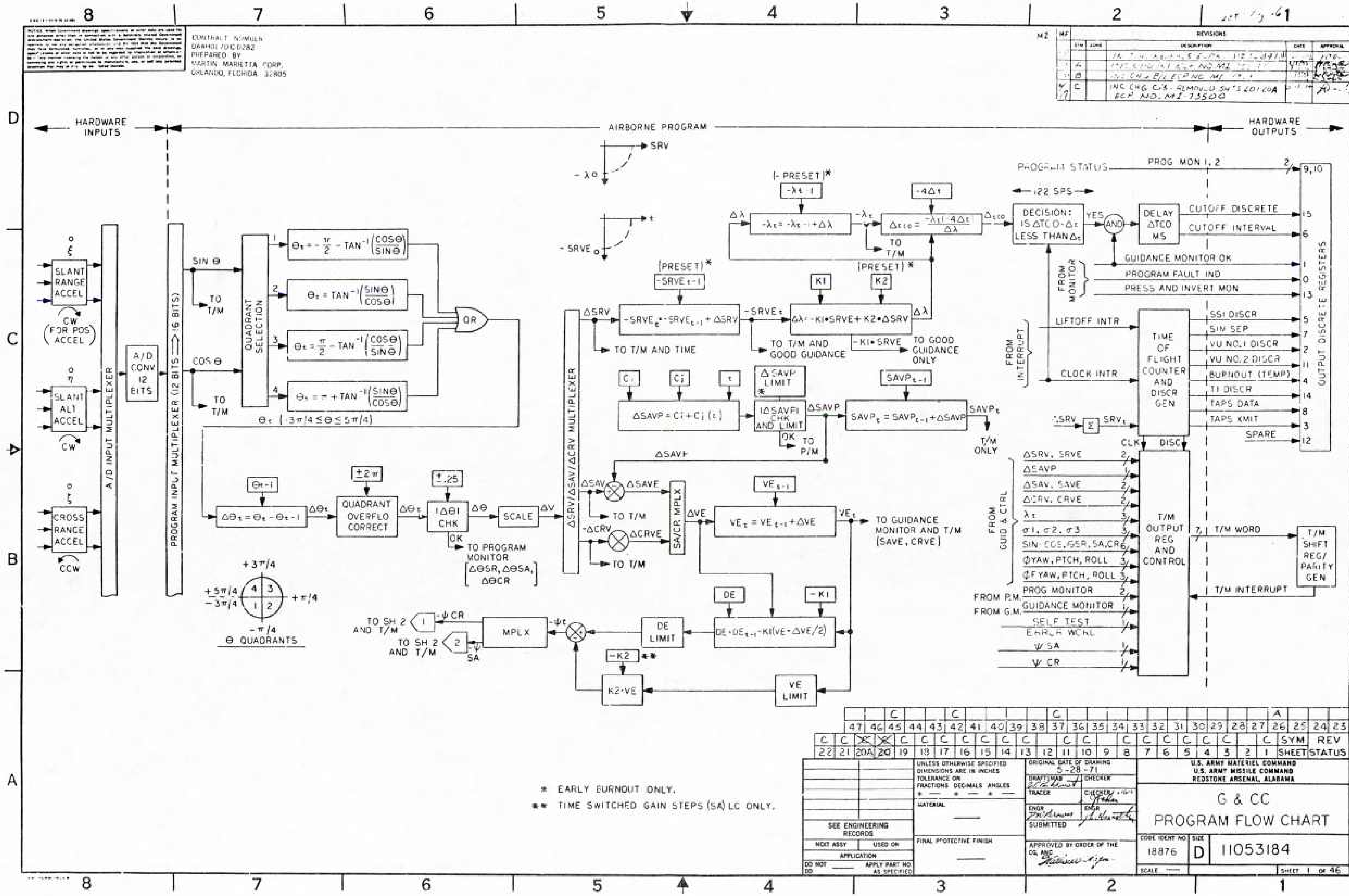


Fig. 3-7 G&CC Program Flow Chart: Sheet 1

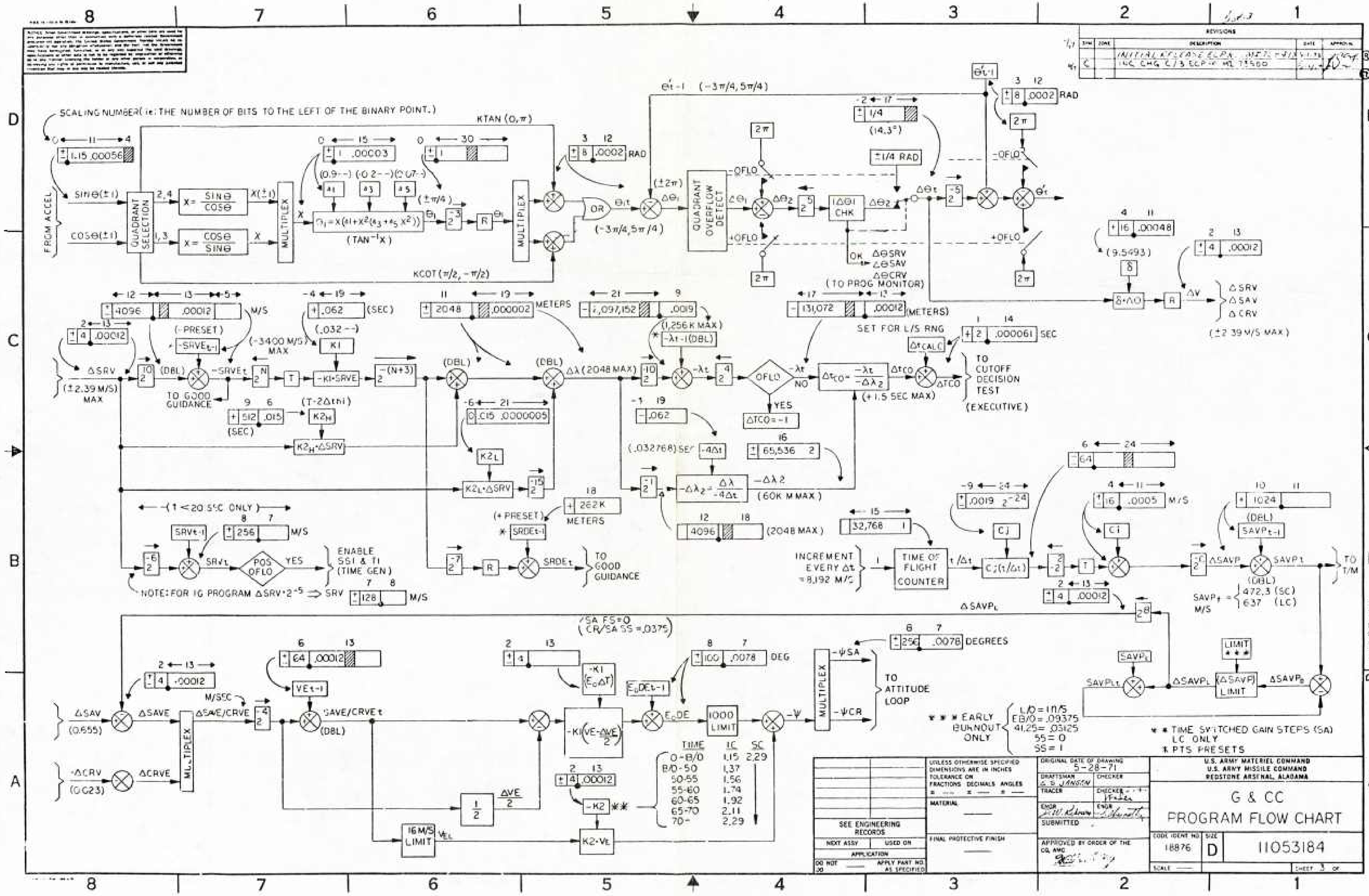


Fig. 3-8 G&CC Program Flow Chart: Sheet 3

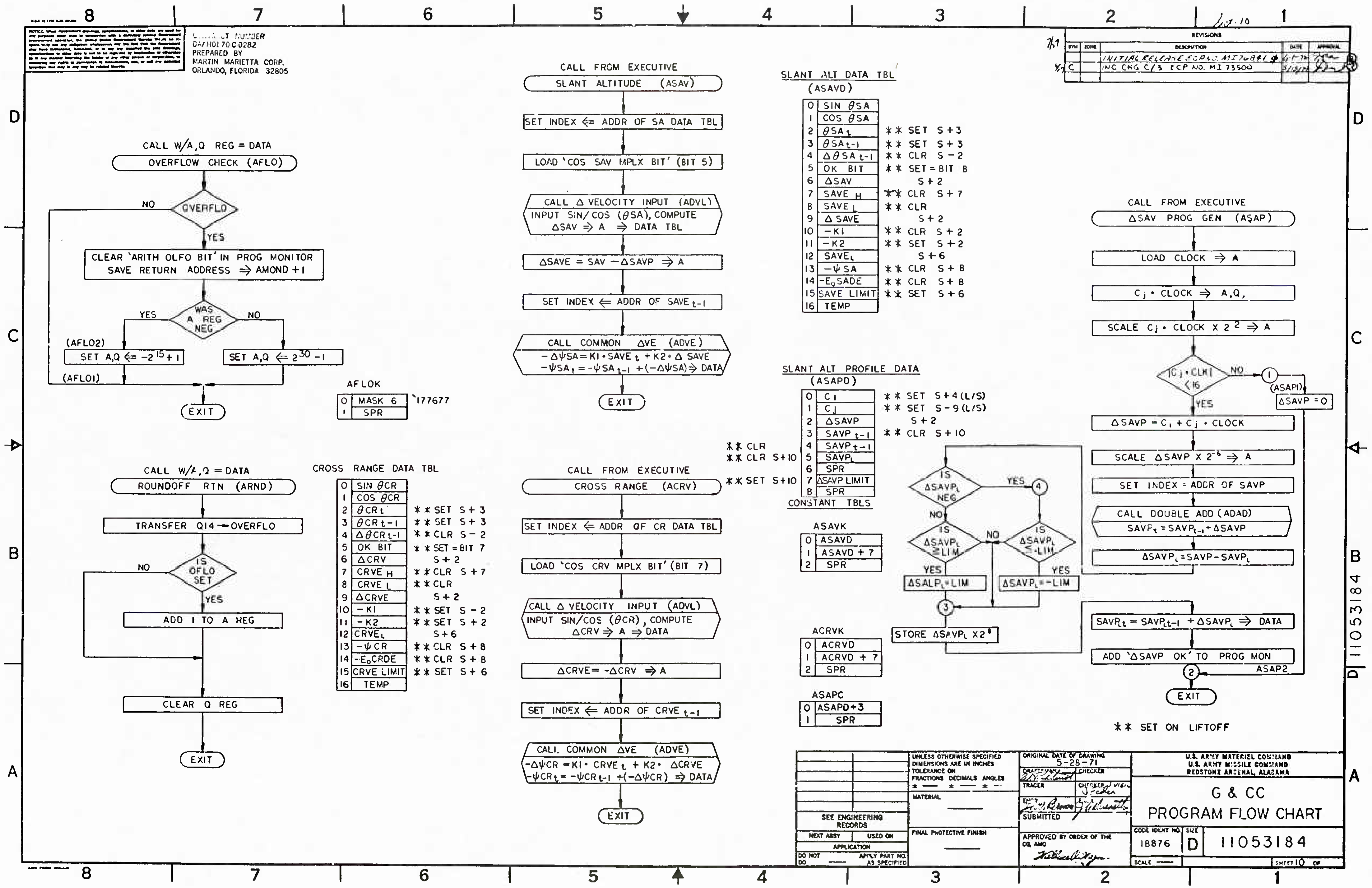


Fig. 3-9 G&CC Detailed Flow Chart: Sheet 10

3.4.3 Software Implementation

The technique used is that of assigning modules to programmers/engineers (note: MMA programmers generally have engineering backgrounds) for detailed scaling, algorithm generation, and flowcharting. Modules are coded and tested, then integrated into the program structure.

All Pershing software programs were coded using assembly language. The primary reasons for assembly language are:

1. PPMO trade-off studies showed that it was cost effective to utilize small core computers if structured programming was not required;

2. Both Pershing computers were off-the-shelf, commercially available, with assembly language compilers already written; and

3. The third generation computers adopted were microprogrammable to expand the basic instruction set from about 40 to over 300 commands.

3.4.4 Hardware Testing Programs

The Type B hardware specifications for the Pershing components or end items contained a fault detection criteria as a function of time. It was left up to the contractors as to how the faults would be detected. For Pershing, MMA tests the items by using software programs in conjunction with the SCTS test equipment. As such, these programs were written without a software specification and according to MMA, under the fixed price contract, would have been prohibitively expensive if documented as the critical programs were. However, documentation for use by technical agencies has proven adequate for the remaining programs.

3.5 SOFTWARE VALIDATION AND INTEGRATION

As indicated earlier in Section 3.4, the Pershing software approach distinguishes between two critical or "prime" software programs and a second group of programs that are less vital. Thus, software validation receives, for this deployed system, major emphasis in the area of the prime tapes used in the field for launching the missile (fire control) and the airborne computer system required for missile guidance. In these cases government standards and specifications for validation, integration, and documentation control are followed by the Pershing Project Manager's Office (PPMO) and the prime contractor, Martin Marietta Aerospace (MMA). There remain over 100 additional software items (field maintenance preassembly checkout, component tests, etc.) which are developed by the same techniques but are not checked, documented, and approved under the same rigid and rather costly procedures used for the prime tapes.

3.5.1 Validation Tools and Techniques

The path through validation includes simulation runs and hybrid laboratory and facilities that include hardware and actual tactical equipment in the deployed configuration. Figure 3-10 is the MMA Hybrid Lab and Hardware interaction configuration used for development of the recent Guidance and Control Computer software. Note that the actual airborne computer and vane control hardware is available in the test lab which is coupled to the Hybrid Lab which, in turn, has the capability of using 6-DOF programs and sensor platforms for many runs through a wide range of ground (PTS) and flight conditions. The Hybrid Simulation is used as a design tool wherein the software specifications are tested and modified as needed. It also acts as an integration tool by bringing together the major software interfaces (Fire Control and Airborne programs) by means of realistic simulation. Finally the Hybrid acts as a software/hardware validation device prior to final flight tests, which are required prior to deployment in Europe of any prime tactical tape. Besides the prime contractor, MMA, the PPMO also uses his own staff and other government facilities (e.g., Inertial Guidance Lab at Huntsville) and other technical sources to validate the software during development through final approval. These supporting activities also have important hardware and simulations available for carrying out their responsibilities.

3.5.2 Software Integration and Fielding

Field integration of the prime Pershing software tapes follows the previously described engineering and checkout phases. The control of the tape during the field integration process is shown by Fig. 3-11, which represents the MMA flow chart from engineering development (Green Tape) through Gold Master and Blue field tapes. Throughout this cycle PPMO and MMA have been coordinating each phase of the software development through technical review boards (see Fig. 3-12 in Section 3.6). The final delivery to the field of the latest tactical tape is a manual process, whereby a field maintenance team will hand deliver the new tape and physically remove the old tape from the firing battery. The modification team will also help in checking out the new configuration.

The close control and high quality of program tapes is the result of the fortunate situation of having the same Program Manager's Office and Contractor groups available to perfect the technique of software development, validation, and field integration. This continuous close association for extended periods of time may not be a universal case in major system software development.

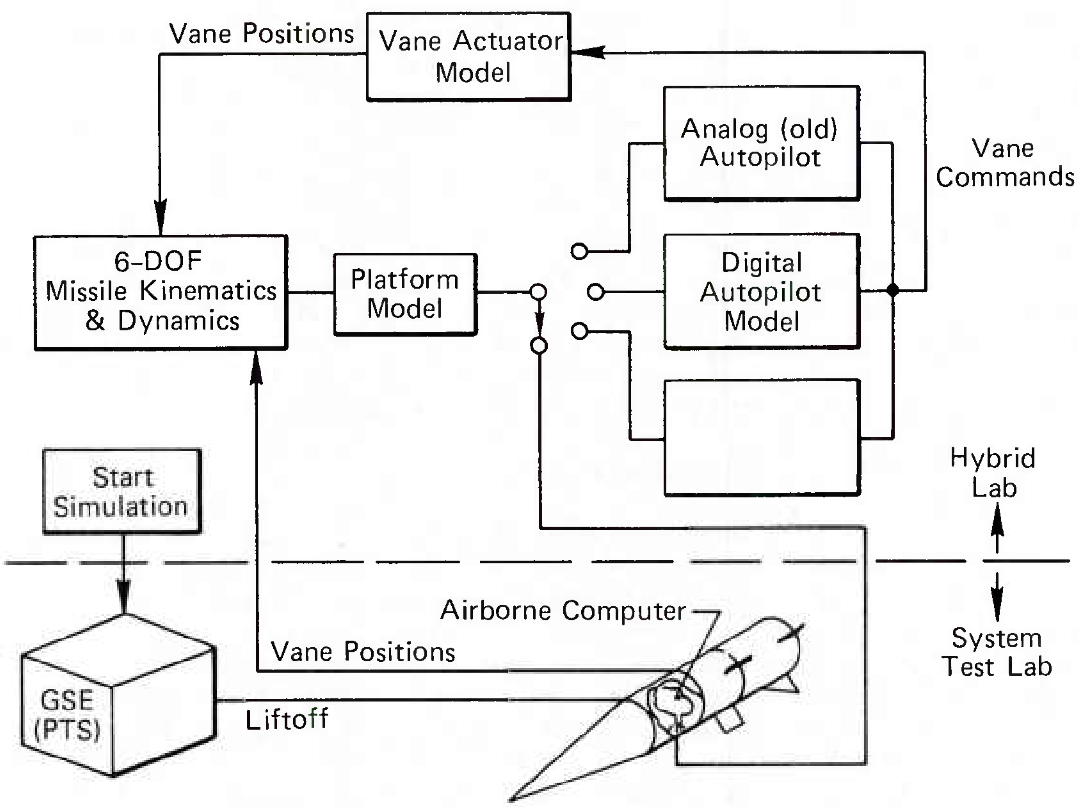


Fig. 3-10 Pershing Hybrid Lab/Hardware Interaction

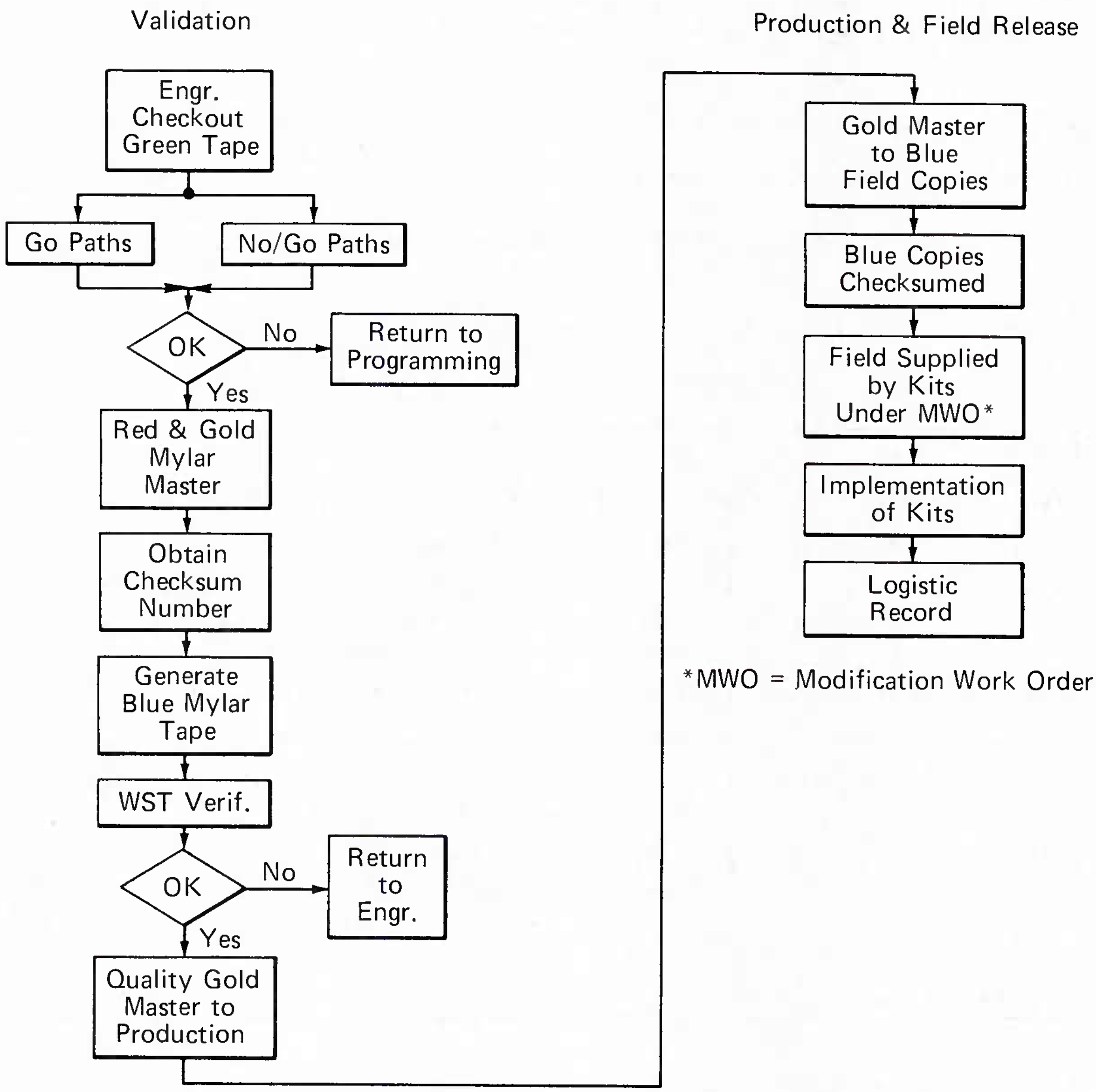


Fig. 3-11 Field Integration Process

The cost aspects of the prime software were mentioned in opening paragraph of this section (3.5). It is difficult to separate hardware design, software development, and validation costs in a program where software is not generally separated from overall engineering costs. However, discussions with knowledgeable Pershing personnel indicate that validation exceeds programming costs by 6:1 (e.g., the approximate cost cited for the G&CC Airborne Program was $1,200,000 for validation and $200,000 for programming and documentation).

3.6 SOFTWARE ACQUISITION MANAGEMENT ORGANIZATION AND METHODS

3.6.1 General Management Information

Pershing management information is summarized in Table 3-5.

TABLE 3-5

PERSHING MANAGEMENT INFORMATION

Program Status	Deployed
Program Manager	PPMO at MICOM, Redstone Arsenal, Ala.
System Contractor	Martin Marietta Aerospace (MMA)
Type Contract	Various over the years (CPFF, Engineering Support by Assigned Task), Maintenance (O&MA), R&D (CPFF) (CPIF), Production (Fixed Price)
Software Contractor	MMA - part of engineering task
Maintenance Agent	MMA - part of engineering support task
Software Deliverables	Tape, specifications, documentation (flow charts, logic diagrams, etc.)
Validation Agent	PPMO Staff, Inertial Guidance Lab (IGL - Redstone), Change Contrcl Board (CCB - PPMO, IGL and MMA Members). Program Manager has overall control
Integration Agent	MMA with CCB and PPMO control

3.6.2 Management Organization

The software management organization has been fairly stable since initial deployment of the system. Briefly, the organization can be described as follows:

1. Program Office: The Pershing Program Manager's Office (PPMO) is headed by a technically competent senior Army colonel with a civilian deputy. There is a staff of engineers and administrative people who assist in the work of this office, which retains strong control of the research, development, maintenance, and supply of Pershing material, including software, operational control of the deployed Pershing battalions are under CINCEUR and CONUS commands but close liaison is maintained with the deployed forces;

2. System Contractor(s): Martin Marietta Aerospace (MMA) is the prime contractor. MMA is given an engineering assist contract and is the primary producer of Pershing software used in the field.

3. Government Laboratories: The PPMO has available government facilities and technical personnel to assist it in the engineering, validation, and documentation aspects of Pershing software. This activity is quite extensive and serves as a strong base for quality assurance of contractor developed hardware and software.

4. Government Maintenance Facilities: In the software area the PPMO and the prime contractor are the maintenance activities. In general, the MICOM RDE Laboratories are used, on a level of effort basis, to maintain the necessary facilities and personnel for a quick reaction capability to perform tasks in the development, validation, and maintenance of contractor software output.

3.6.3 Management Techniques

The DoD Configuration Management Program (5010.19 and 5010.21) is generally followed by PPMO for software activity. The MIL-STD-490, Type B-5 specification has been used as establishing the base line for Pershing software. Changes thereafter are controlled by Pershing Engineering Change Procedures (PECP) (Subsection 3.6.4) in accordance with MIL-STD-480 Specification that in turn generates a Mandatory Item Specification (MIS) technique. The Pershing Specification Tree, Fig. 3-6, indicates the different MIS's used for management of software design and documentation.

3.6.4 Pershing Engineering Change Procedures (PECP)

As Pershing has been deployed for over ten years, current software development and fielding is considered an engineering procedure that primarily involves changes to existing programs (or the incorporation of a new subprogram into the existing system). The Pershing Engineering Change Procedures (PECP) have been developed to formally control this process. Figure 3-12 is the contractor's (MMA) view of his responsibilities and the controls exerted by PPMO and his supporting technical facilities. The Configuration Control Board (CCB) and the Change Planning Board (CPB) are important control groups wherein the prime contractor and PPMO make the basic decisions on the "start, stop and go" of software development. These boards are normally chaired by the government and have both PPMO and MMA representation. Following the Fig. 3-12 flow, it could take as long as one or two years to field a software change for the prime programs.

3.6.5 Documentation Requirements

APL is a user of the documentation identifying the deployed software in our work of system evaluation. We have found the various specification, design, operational, test, and record documents as being accurate and complete. Figure 3-13 illustrates the steps involved in the development of documentation for the Ground Support Equipment software and is representative of the documentation requirements used for the major Pershing software items.

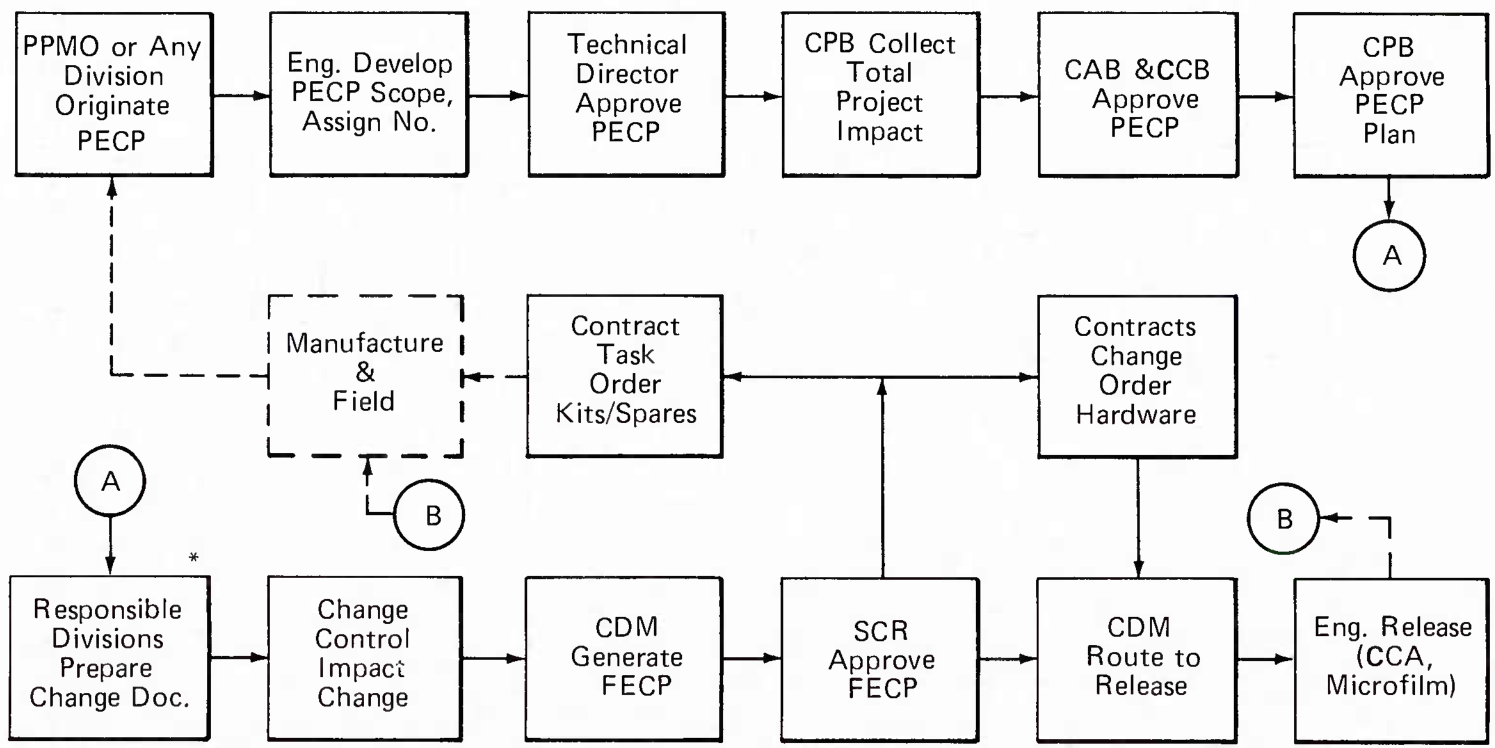
3.7 OPERATIONAL SOFTWARE MAINTENANCE

The PTS provides on-site basic test and malfunction analysis and the SCTS is used to perform near area maintenance and program checking.

No changes are made to the "sealed" tactical software tapes in the field. As indicated earlier, detailed and stringent procedures are followed to validate and certify the prime tapes. Any changes or corrections would follow the Pershing Engineering Change Procedures (Subsection 3.6.4). MMA and the PPMO are the maintenance agents with the MICOM RDE Laboratories providing all necessary facilities for software maintenance.

3.8 HIGHLIGHTS

An example of the PPMO technical control of computer hardware is the recent digital "Airborne Tactical Computer" development program. PPMO set the upper limit on the amount of core at 3200 words. When requirements forced the total over this number, tradeoffs were visible to the PPMO who then made the final decisions on what was to be cut back.
(MP1, SE1)



*Liaision Call Flow Starts Here

PECP: Pershing Engineering Change Proposal
CCB: Configuration Control Board
CPB: Change Planning Board
CDM: Change Data Management
SCR: Senior Command Representative
FECP: Field Engineering Change Program

Fig. 3-12 PECP Change Flow

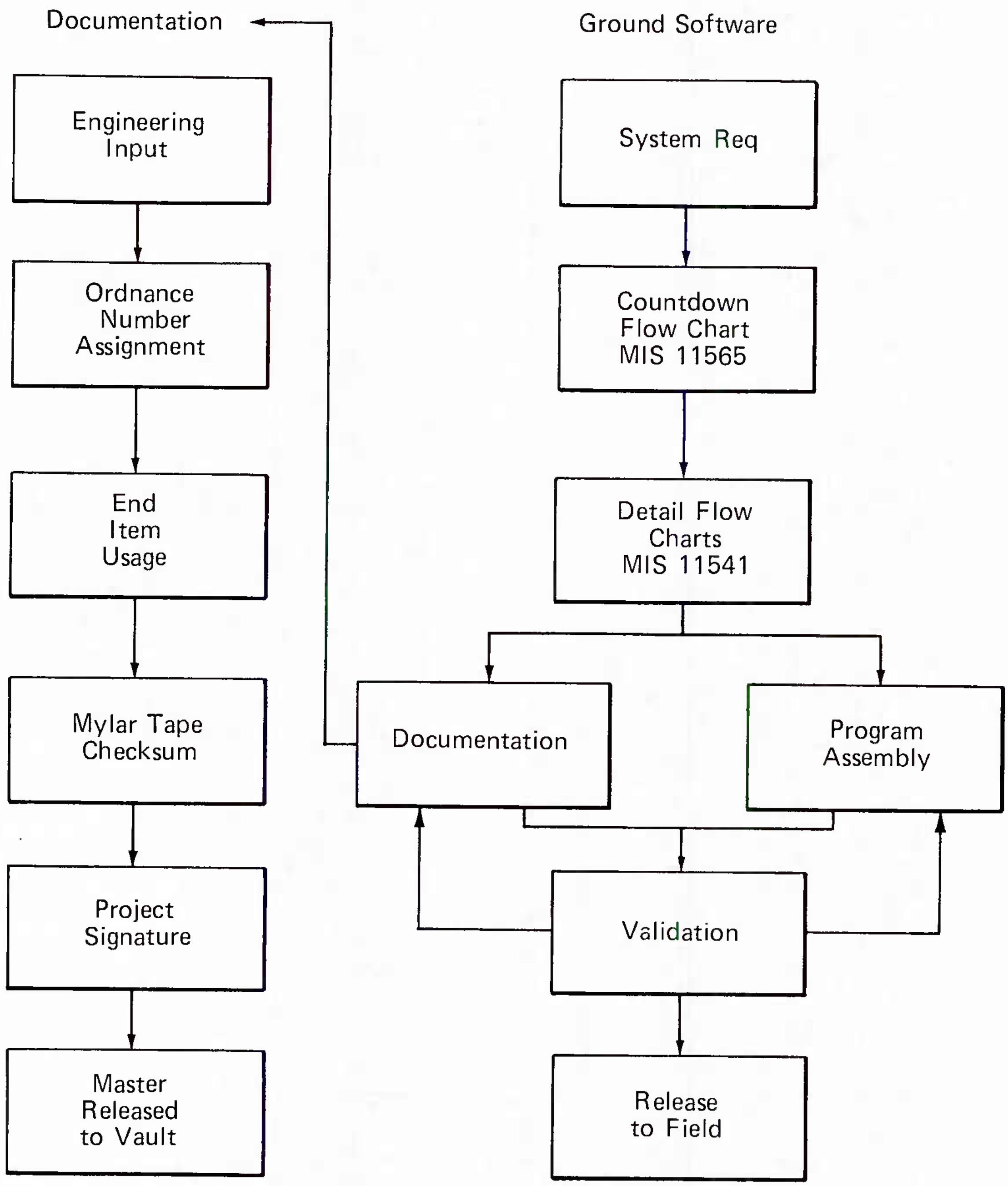


Fig. 3-13 Development of Ground Support Software Documentation

The contractor does not use, in any important way, personnel designated as programmers. Software is designed, developed, validated, and maintained by engineers who have the capability of programming (IP2)

Two Pershing software programs (Airborne and Countdown) are strongly controlled by PPMO who uses the DoD Configuration Management Program established by DoD Directive 5010.19 and DoD Instruction 5010.21. Specifically, they set forth the requirement to use MIL-STD-490, Type B-5, specification to assure development of a computer program satisfactory for the intended use. After the baseline is established using MIL-STD-490, changes can only occur using guidelines in MIL-STD-480. This technique also generates a Mandatory Item Specification (MIS) as required by ASPR 1-202(a). PPMO elected to have only the two most critical of the over 149 software programs controlled by the above technique because of the costs involved in implementing the government standards. (AM1, MP3)

The PPMO has had a strong hand in Pershing hardware and software design, quality assurance, and maintenance. It has been able to keep this control by having an engineering staff available to it at MICOM (Redstone Arsenal). (MS1)

The PPMO has used computer software contractors as advisors on management techniques. (MS1)

The contractor indicated that in the critical programs, the verification costs exceeded the programming costs by a 6:1 ratio. (TT1)

No attempt was made in Pershing to standardize computer hardware or languages used.

4. SAM-D WEAPON SYSTEM

4.1 GENERAL SYSTEM DESCRIPTION

SAM-D is an air defense weapon system designed for field use by the U.S. Army. It is an advanced surface-to-air guided missile system, able to operate in an ECM environment with a high single-shot kill probability (SSKP) and the capability to conduct multiple, simultaneous engagements against high performance targets of the 1980-1990 time frame.

SAM-D is designed for rapid strategic deployment and tactical movement (including air transport, airborne operations, sea transport in cargo ships and landing vessels, rail transport, and ground mobility on wheeled vehicles).

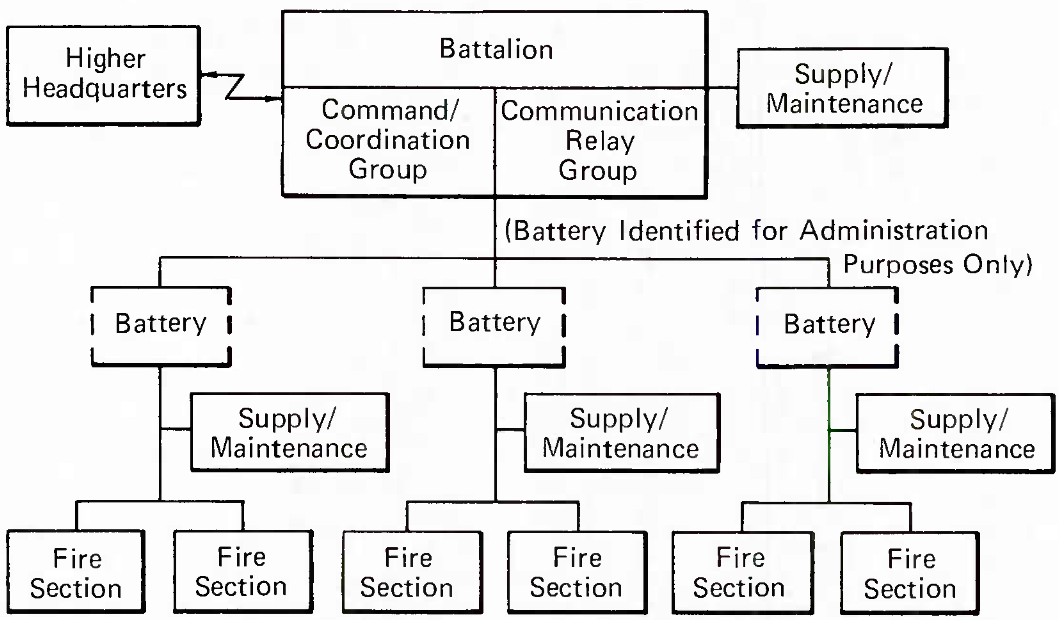
According to its requirements document, SAM-D must detect, identify, engage, and destroy high performance aircraft and missiles. It is a medium range weapon system intended to be used with short range air defense systems of the Army in the field and will complement the Air Force in the overall air defense in the theater.

The system has two operational echelons - Battalion Control and Fire Control Section. The Battalion level commander exercises operational control over and coordination of the Fire Control Section operations including rules of engagement, operational procedures, and management direction. The Fire Control Section performs all functions associated with the detection, tracking, identification, engagement, and destruction of targets. Top level Air Defense planning and direction are external to the SAM-D system and interface at the Battalion level. Figure 4-1 shows the overall Battalion organization.

Significant SAM-D features include modular system organization (variable force level), decentralized operations (multiple simultaneous engagement capability), digital data processing (traffic handling capacity), advanced technology radar missile and guidance (target resolution, multiple simultaneous engagements, and penetration tactics resistant), inherent and integral ECCM, minimum number of major items of equipment, new support concept (operational availability), and small number of tactical vehicles (strategic and tactical mobility).

4.1.1 SAM-D Subsystems

At the Battalion level, the major items of operational equipment are the Command and Coordination Group and the Communications Relay Group. (Up to seven of these per Battalion may be located throughout



Fire Section (FS) Consists of:

(Up to 6 FS in Each Battalion)

Fire Control Group – 1 Each
Launcher Group – Up to 8 Per Fire Section
Missile Round–Minimum of 20 Ready Rounds Per Fire Section

(10/1/72)

Fig. 4-1 SAM-D Battalion Organization

the defended area to provide communication links among the various Weapons Control Groups and the Command and Coordination Group as required by the deployment and distribution of Fire Sections).

At the Fire Section level, the major items of operational equipment include Weapons Control Group (one), Radar Group (one), Launcher Group (up to eight), Missile Round (four per launcher group), and Prime Power Group (one).

The Radar Group consists of the Radar Unit mounted on a suitable vehicle, including a phased array antenna and a waveform agile transmitter. The beam forming capability of the antenna allows the beam to change its position within a time interval of 20 μs, and the transmitter can generate any of several waveforms at each beam position.

The Weapon Control Group is also vehicle-mounted and consists of the Weapon Control Computer and the appropriate software.

The Launcher Group consists of a launcher unit and an electrical power unit on a suitable vehicle. Four ready missiles in separate cannisters are carried on the launcher group. The launch bed elevates and turns in azimuth prior to missile launch. The launcher unit serves as an interface unit between the Weapon Control Group and the ready missiles and performs those functions necessary to prepare and launch missiles as directed by the Weapon Control Group. The missile round fires out of its cannister and requires no pretesting or launch site maintenance.

The Prime Power Group has four 60 kW generators, and the fuel supply and is mounted on a suitable vehicle. It supplies all of the electrical power to run the units of the Radar Group.

4.1.2 Acquisition History

SAM-D is the result of more than 15 years of continuing effort (see Table 4-1) by the Army and its contractors to establish and produce an air defense system for use with the Field Army. SAM-D is a descendant of the Field Army Ballistic Missile Defense System (FABMDS), which was intended primarily for defense against ballistic missiles. After more than two years of study, the FABMDS program was terminated in October 1962 when the Army decided it was too complex and costly.

Almost concurrent with this action, the Army's Combat Development Command (CDC) concluded a comprehensive study for air defense of Allied Command Europe that proposed the development of an air defense system called Improved Surface-to-Air Missile to replace Hercules and Hawk. This system, less complex than FABMDS, was to be optimized against the air supported threat of the 1970's and have an inherent Anti-Tactical Ballistic Missile (ATBM) capability. Technical feasibility studies for the system (AADS-70) were conducted in 1963 and 1964. In October 1964, concern was expressed at the Office of the Secretary of Defense (OSD) level about the particular ballistic missile threat that AADS-70 should handle. It was believed that a properly designed ground air defense missile system could be provided with some explicit ATBM capability. Accordingly, the program was reoriented at that time to a combined component verification program and a study of tradeoffs on which to base the design of the future system.

TABLE 4-1

SAM-D PROGRAM HISTORY

1959	Military Requirements Study Review
1960-1965	Concept Formulation Studies
1966	Contract Definition
1967-1971	Advanced Development
1971	Engineering Development Definition
1972	Authorization of Vulnerability Study
1972-1974	Engineering Development
1974-1975	DSARC II Review and Recycle to Transition Phase to Develop Validation Data (Proof-of-Principle)
1975	DSARC IIA Review
1976 ?	Resume Engineering Development
1977-1988 ?	Production and Deployment

The SAM-D project was established in August 1965 as a result of these reorientation studies, and in March 1966 OSD directed the Army to proceed toward advanced development. The Request for Proposal (RFP) for contract definition was prepared and issued to 13 qualified bidders in April 1966. A major objective in the RFP was the provision of an integrated, but flexible, building block approach for an air defense system. Three of the responding contractors, Hughes, RCA, and Raytheon were selected for fixed price contracts to complete contract definition of their proposals. These contractors were required to conduct tradeoff, optimization, and other definition studies necessary to verify recommended technical approaches; establish firm and realistic schedules on cost estimates for development; and to plan estimates for subsequent production engineering, facilities construction, and production hardware. Planning schedule and cost estimates were required for program life cycle phases including operation and maintenance.

In May 1967, the Raytheon Company was selected to proceed with Advanced Development for SAM-D. In November 1967, the initial letter contract was completed as a cost-plus-award-fee contract for a period of 28 months.

In 1970, it was decided that prior to entering Engineering Development, another contract definition effort should be undertaken to define the work to be done under the Engineering Development contract. Since contract definition had last been performed in 1966, a new, essentially similar, program called Engineering Development Definition was instituted. This program was completed in 1971.

At the same time, the Army Chief of Staff established an Air Defense Evaluation Board to review the capabilities of U.S. Army Air Defense Weapon Systems. The ADEB investigation covered the threat, weapon system performance, cost effectiveness, and risks associated with the SAM-D program. As a result of the ADEB investigation, certain threat characteristics and SAM-D performance requirements were changed, and these changes were projected into the Engineering Development Definition process as the system definition evolved. The investigation resulted in recommendations that SAM-D enter engineering development in FY-70 and that SAM-D development maintain the option to add a nuclear warhead and antimissile capability.

The SAM-D Defense System Acquisition Review Council (DSARC) reviewed SAM-D on 4 February 1972. This review was supported by the SAM-D revised Development Concept Paper (DCP) No. 50. The SAM-D DCP was approved in March 1972, and a contract for engineering development was executed with Raytheon Company.

In the approval of DCP No. 50, directions were included to:

1. Develop options to remove marginal features and associated costs; e.g., removal or reduction of nuclear hardening requirements except those associated with electromagnetic pulse (EMP), the removal of one transmitter train or reduction in power of both, removal of sidelobe cancellers, and simplification of the Radar Receiver and Weapon Control Computer;

2. Provide another appropriate contractor for at least one year to work on an intensified effort examining possible changes of specifications and design and alternate solutions that would result in a cost reduction; and

3. Give serious consideration to using an incentive fee provision to provide additional emphasis on contractor management performance.

In early 1974 a DSARC II review was conducted. At this review it was decided that the SAM-D system would enter a transition phase to develop validation data particularly for the Track-Via-Missile principle. Currently, the validation tests are being conducted at White Sands and a DSARC IIA review is tentatively scheduled for late 1975. Concurrently, work on the essential features of the engineering development single fire section model of SAM-D is proceeding at a reduced level.

SAM-D is currently in the Advanced Development Phase with emphasis on demonstration of guidance feasibility. Computers and computer programming structure are being developed to meet the specific needs of the SAM-D system.

4.1.3 System Block Diagram

Figure 4-2 shows the relationship of the data processing system of SAM-D to other elements of a SAM-D Fire Section. The Weapon Control Computer and peripheral equipment are the major data processing elements of the SAM-D system in the Weapon Control Unit. The Radar Unit and the Launcher Unit contain digital signal processing and control elements. These are hard wired elements and are designated "firmware". The firmware elements will not be treated in detail in this report.

4.2 COMPUTER SYSTEM ARCHITECTURE

4.2.1 Weapon Control Computer (WCC) General Description

The Weapon Control Computer (WCC) is a multiprocessor with growth capability in its storage capacity, processing ability, and input/output handling. Figure 4-3 shows the major components of the WCC in block diagram form. WCC functions and computer characteristics are summarized in Table 4-2.

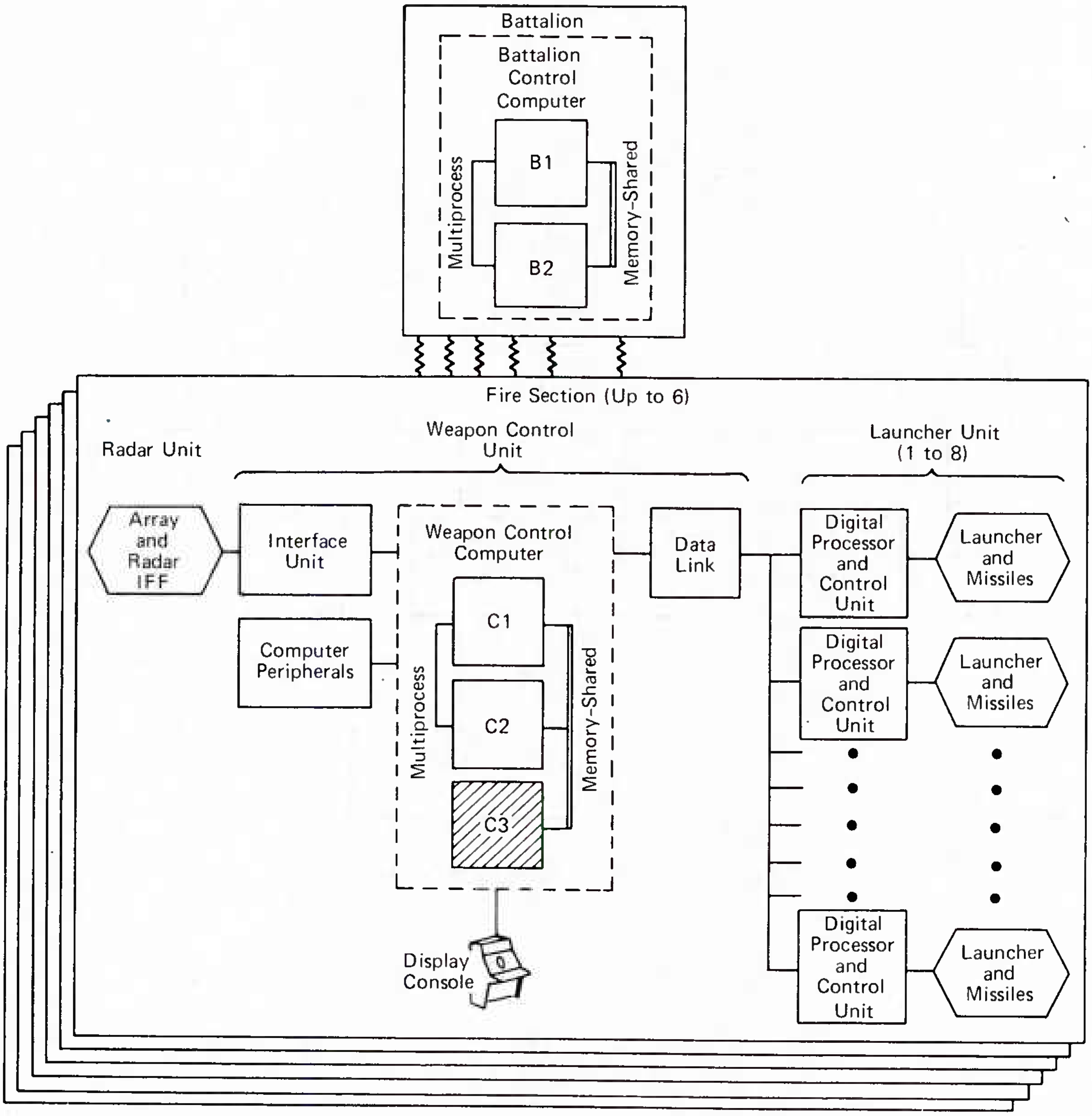


Fig. 4-2 SAM-D Weapon System

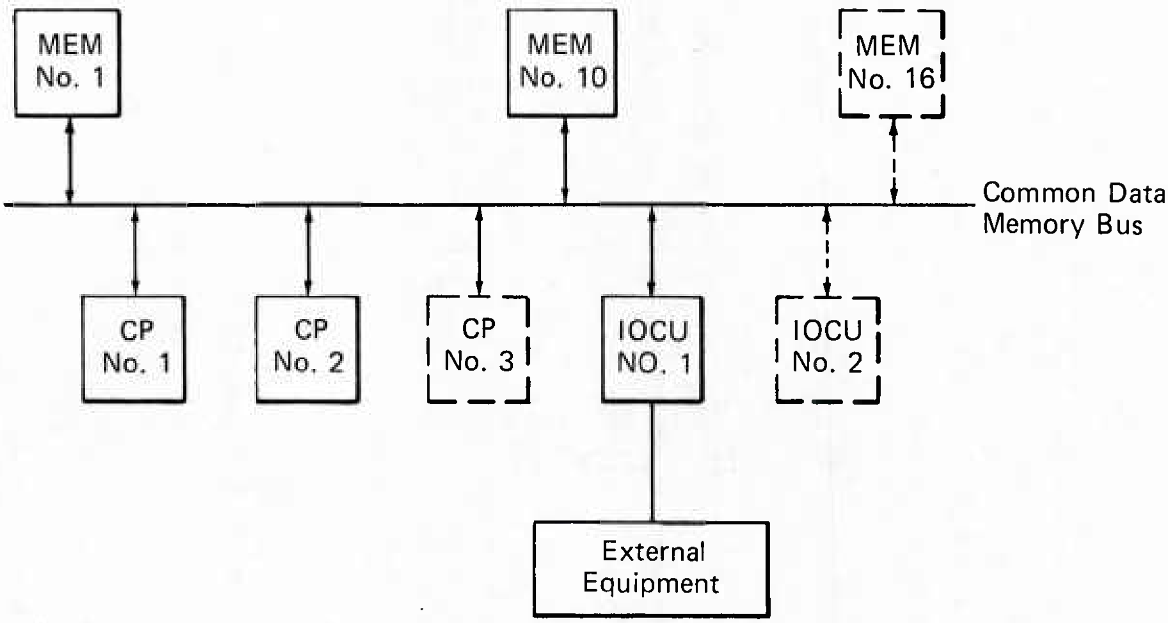


Fig. 4-3 WCC Major Elements

TABLE 4-2

SAM-D COMPUTER SUMMARY

Unit	Type	Function	Processors	Memory
C1, C2	Raytheon (24-bit, 1 μs)	Weapon Control Computer (multiprocessor); radar control, detection and track, communications, display, weapon assignment, status monitoring, missile guidance	2	160k (expandable to 256k
C3	Raytheon (24-bit, 1 μs)	Future third processor	(1)	

Table 4-3 lists the WCC characteristics that are a result of numerous functional analyses and tradeoff studies. The computational rate and storage required, including data and programs, were determined by assuming worst-case scenarios. To a large extent, extrapolations were used from the actual Advanced Development (1967-1970) operational software to translate the postulated threat and environment into data processing terms. Thus, once the computer "horsepower" and storage requirements were established, the other characteristics could be determined from the tradeoff studies.

TABLE 4-3

WCC CHARACTERISTICS

Configuration	Multiprocessor
No. of Central Processors (CPU)	Up to 3
Computational Rate*	10^6 Equivalent Adds/Sec per CPU
Storage Capacity	256k Words Max
Max I/O Rate	10^6 Words/Sec
Vol. for Max Growth	27 ft^3
No. of I/O Control Units (IOCU)	Up to 2 with 16 Direct Mem. Access Channels per IOCU

*The computational rate is currently under review by Raytheon.

As shown in Fig. 4-3, the major elements of the WCC consist of:

1. Modular on-line memory units (10 expandable to 16);

2. Central processing units (2 expandable to 3);

3. Input-output control unit (1 expandable to 2); and

4. External equipment.

The memory stores the operational programs and data. The central processing units perform the computation and logical decisions. The input-output control unit controls the transfer of data into and out of the WCC. The external equipment provides data and signal conditioning for signals transferred between the WCC and the rest of the Fire Section.

A control and Maintenance Panel (not shown) permits initial loading of operational software or off-line loading of diagnostic programs. It also permits indication of internal WCC status and controls the mode of the WCC; i.e., normal run, halt, single step, and single instruction.

4.2.2 WCC Input-Output Control Unit Interfaces

Figure 4-4 shows the Input-Output Control Unit (IOCU) interfaces between the WCC memory bus and the external equipment. There are five groups of ports. These are allocated as shown. The radar uses the highest number of ports because of the high data rate.

The peripheral equipment is used as part of the tactical hardware configuration for providing:

1. A protected and hardened backup for operational programs and critical constants for use during initialization and nuclear radiation recovery;

2. For initial loading of programs into the hardened mass memory device;

3. For storing time and events recording of mission operations (data logging); and

4. For hardcopy printouts of standard operating procedures, modes, march orders, etc.

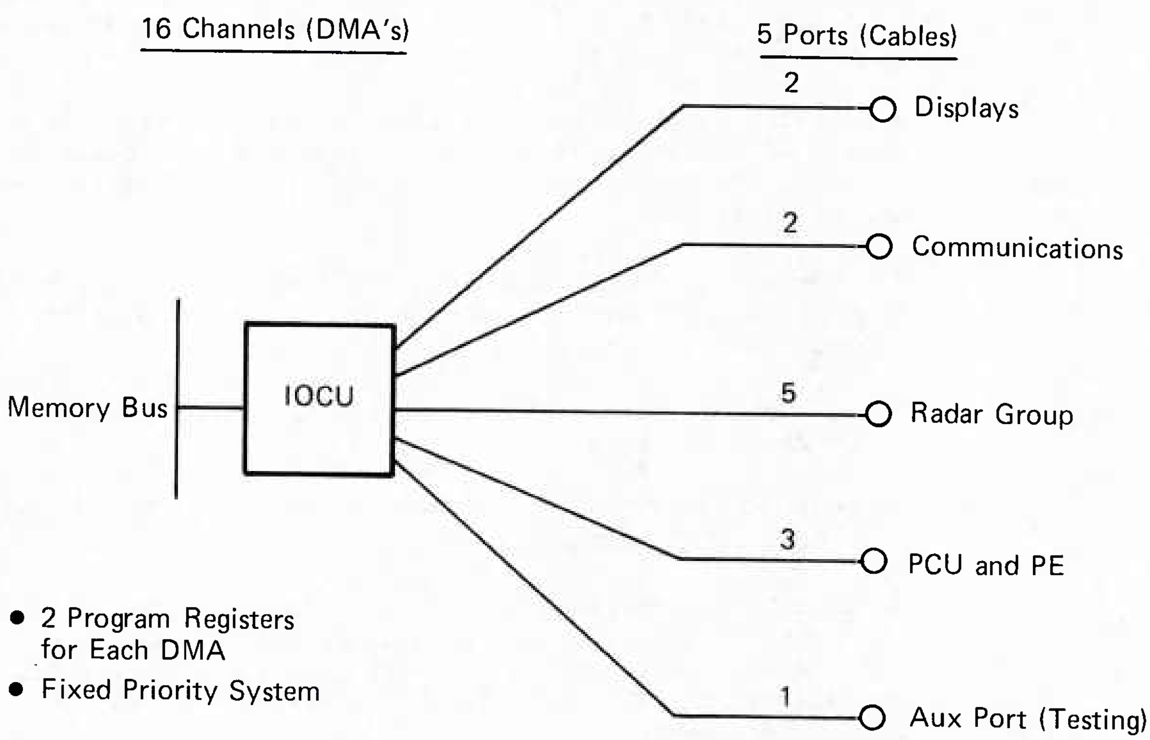


Fig. 4-4 IOCU Interfaces

4.2.3 Computer System Design Rationale

The salient design requirements resulting from the analyses of system requirements were defined to be:

1. 152,000 words of on-line storage expandable to 262,000 words modular memory concept.
2. 1.8 million adds/sec processing capability (2 CPU's).
3. Militarized hardware capable of withstanding the environment imposed on the Fire Control Group and Battalion Control Group and able to meet the availability requirements allocated to the WCC.
4. A computer capable of operating or recovering from the levels of nuclear radiation and electromagnetic pulse imposed on the SAM-D system within the specified system reaction time.
5. The use of standard components, replaceable units, and mechanical structures to as great an extent as possible without compromising the design.
6. A total volume constrained (under maximum configuration) to be 27 ft^3 or less.
7. An instruction set optimized for use with JOVIAL compiler-generated SAM-D computer programs.
8. An interface compatible with the displays and controls, communications data links and multifunction radar, and capable of handling the data traffic rate under peak conditions within all external time constraints (i.e., 10^6 words per second).
9. A means for permitting operator monitoring and manual control of the computer for initialization and debugging operations.

4.2.3.1 <u>Configuration Study</u>

The first major study was performed to establish the optimum configuration for the WCC. Three structures were considered, namely: a uniprocessor configuration, a multicomputer configuration, and a multiprocessor configuration. The multiprocessor architecture was selected since it was considered optimum for SAM-D and presented minimum risk in development. It was further determined that a maximum of three ce processors should be employed to perform the tasks known at the time the study was conducted. Examination of the state of the art in central processors led to the conclusion that a 1.0 million adds-per-second processor was within the realm of feasibility using militarized hardware and hence a two processor multiprocessing configuration would be the least cumbersome and most efficient architecture obtainable with minimum risk.

4.2.3.2 Processor Studies

Having decided on a 1.0 million adds-per-second central processor (CPU) several studies were conducted to determine:

1. How best to achieve the 1.0 million adds-per-second computational rate, and

2. What characteristics and features should the processor assume.

To meet and sustain a 1.0 million adds-per-second rate it was determined that the CPU required a bandwidth of 2×10^6 words/second with memory, which was achievable if each CPU could simultaneously access two memory units. Thus, memory overlap was deemed to be an essential feature of the CPU.

4.2.3.3 Word Length Analysis

An extensive study was performed to determine optimum word length requirements for both data and instructions. Factors considered are listed below:

1. Data accuracy requirements,

2. Memory addressing requirements,

3. Instruction implementation ease,

4. Cost to hardware and software development,

5. Effects of word length on storage requirements and execution rate,

6. Index register requirements, and

7. Programming considerations.

A 24-bit word length was selected as a result of this study based on cost effectiveness as the governing parameter. In parallel with the word length analysis a study was conducted to determine whether floating point hardware should be introduced. Since floating point capability is not unrelated to length, the two studies were combined with the result that the CPU is defined to have both a 24-bit word length and floating point hardware.

4.2.3.4 Instruction Repertoire

In selecting an optimum instruction set, the first step taken was to examine the utilization of the original Advanced Development (AD) instructions. Those instructions that could not be readily used by the AD JOVIAL Compiler became prime candidates for elimination. Infrequently used instructions were also eliminated. Experience gained during the 1967-1970 Advanced Development phase indicated that storage savings and speed enhancement could have been accomplished by use of certain instructions not available in the original Advanced Development WCC. These instructions were added to the modified Advanced Development instruction set in place of the deleted instructions. The original Advanced Development multiprocessor control instruction was modified to give increased capability required for Engineering Development. A test and set instruction on a bit-by-bit basis, a bit/byte manipulation instruction and basic floating point arithmetic instructions were added to provide flexibility and JOVIAL compatibility.

4.2.3.5 Computer Memory Analysis

The following tradeoff studies were conducted:

1. Technology,

2. Core Organizations, and

3. Module Size

The first of these tradeoffs looked at the speed requirements for compatibility with a 1.0 million adds-per-second central processor and volume constraints. The technologies investigated consisted of ferrite core, solid state (bipolar and MOS), plated wire and thin film. Early in the study, plated wire and thin film were eliminated, leaving core and solid state as the main contending technologies. Core was finally selected, based primarily on minimum development risk and past experience. A further tradeoff examined 3D, 2 1/2D and 2D core organizations and concluded that a 2 1/2D 3-wire system would be optimum for the WCC.

In considering memory module size, upper and lower bounds were established resulting in a tradeoff between 8k, 16k and 32k module sizes.

The 16k size was selected as exhibiting the best overall features of the many parameters weighted in making the evaluation. Among the parameters considered were:

1. Cost,
2. Volume,
3. Reliability,
4. Processor conflicts, and
5. Internal bus complexity

4.2.3.6 Computer Technology Study

A survey of military computers available at the time of the study (1971) or in the process of being developed was conducted, manufacturers were contacted, and the facilities of the top contenders were visited for more in-depth interviews and information gathering. Thirty-five manufacturers and their product lines were evaluated against broad SAM-D baseline requirements, and the number of candidate computers were filtered down to four. None of these four machines completely met all of the SAM-D baseline requirements.

A second aspect of the technology study examined the state-of-the-art in single processor techniques with a view toward evaluating such techniques for applicability to the Engineering Development WCC. A number of very high-speed commercial processors were studied and techniques used for achieving their advertised speed were evaluated against cost and risk in militarized hardware implementation.

The Computer Technology Study effort was supplemented by contracting with Auerbach Corporation to conduct the vendor surveys and participate in product line evaluations to satisfy the SAM-D requirements.

4.2.3.7 Packaging and Cooling

An overall SAM-D packaging study was performed during Engineering Development as part of the Standardization studies. In parallel with this effort, the problems associated specifically with the WCC and the constraints imposed upon it by the Weapon Control Unit (WCU) were investigated as part of the Data Processing System Definition studies. It was concluded that the standard mechanical structure approach generated out of the Standardization studies with minor modifications (dimensional rather than conceptual), would permit the WCC to be packaged within the 27 ft^3 constraint.

A second study was conducted to determine the cooling requirements and techniques to be employed. It was concluded that air cooling would be best for the WCC in light of the stringent volume restriction imposed and from an economic point of view.

4.3 COMPUTER PROGRAM ARCHITECTURE

Figure 4-5 shows the general allocation of Fire Section functions by the WCC.

The Tactical Software Program Architecture consists of the following five parts:

1. Operational Executive,
2. Surveillance Software,
3. Command and Control Software,
4. Guidance Software, and
5. Emplacement Software.

4.3.1 Operational Executive

The ensemble of applications programs operate under the control of an Executive Program. These programs must be organized and implemented in order to perform their functions with minimum dependence on operator intervention. Operator involvement is required only to the extent of the mission requirements for operator decisions. Thus the Executive Program must control use of data tables, sequence programs according to a priority structure, and respond to interrupts.

4.3.2 Surveillance Software

Surveillance Software performs the surveillance function, i.e., the detection and tracking of real targets in the search volume, in direct conjunction with the radar unit. The scheduling software must provide the facility of driving the radar in conformity with system requirements and limitations. The radar message formatting software must format radar action requests for radar unit use, while the radar return message processing software must act on the various returns to initialize or update various tables in the data base. Other software is required to ensure that search returns from targets already under track do not initiate duplicate track files, to predict target position for track action requests, and to control or change search sector boundaries. In summary, these

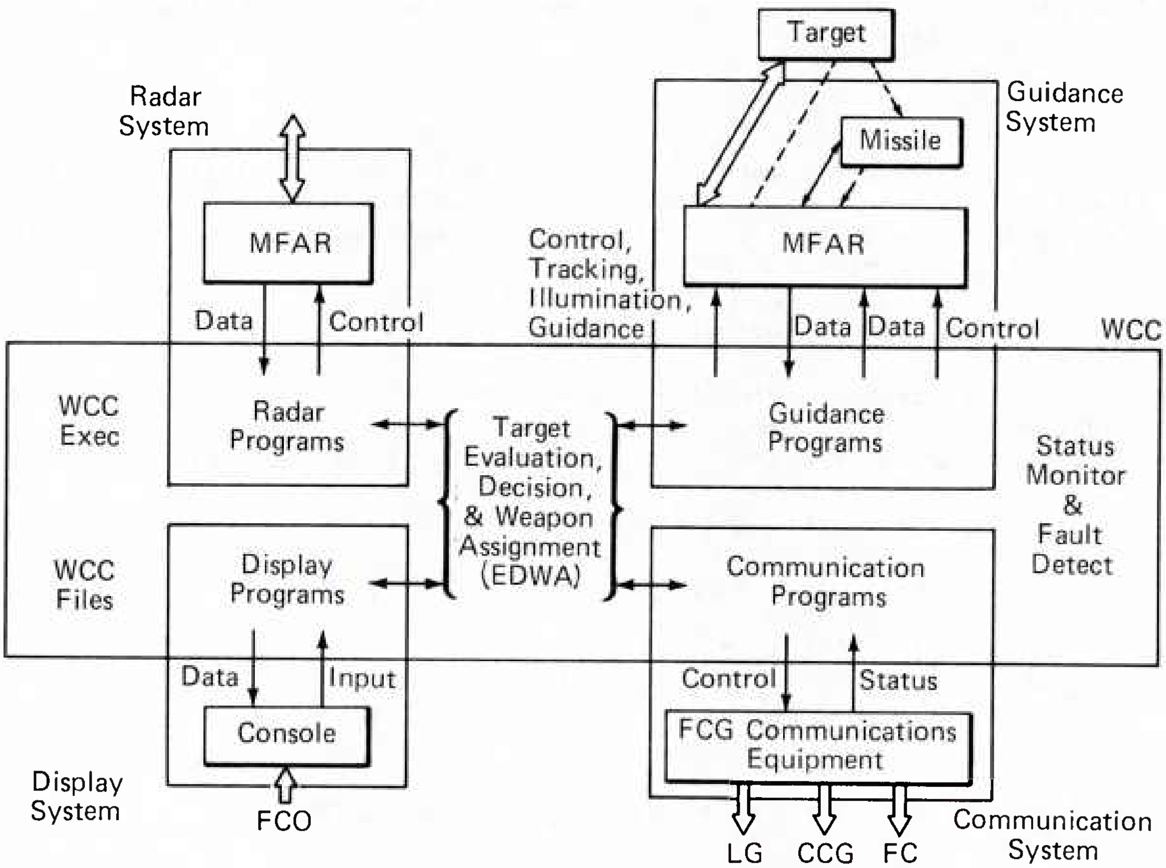


Fig. 4-5 Control of Fire Section Functions

programs provide the estimates of position and velocity (in target files) required for both the surveillance function itself and the other functions of the system.

4.3.3 Command and Control Software

The command and control function is subdivided into four parts: Communications, Display, Engagement Decision and Weapon Assignment (EDWA), and Status Monitoring.

4.3.4 Guidance Software

Guidance software is required to control and guide the missile flight through the several phases of acquisition, midcourse, TVM acquisition, and TVM track, and also to process the various guidance action signals in either benign or ECM environments.

4.3.5 Emplacement Software

The need to support Emplacement and System Initialization arises every time the SAM-D system is moved. To support this requirement, supplemental programs must be furnished to do the following:

1. Generate Clutter Maps,
2. Initialize Radar Programs,
3. Initialize EDWA, and
4. Support Operator/Computer Communications.

4.4 SOFTWARE DEFINITION, DESIGN, AND IMPLEMENTATION

4.4.1 Software Definition

4.4.1.1 Planning

Software development proceeds in discrete steps in which software activities are identified consistent with system and hardware development steps. Initially, system performance requirements are transformed into data processing system requirements. More detailed analysis produces documentation defining design requirements. Parallel efforts establishing test criteria, requirements, plans, and procedures are conducted. Program units and packages are then coded, tested, and released for operation usage. The items resulting from these activities lead to the establishment of the contractor product baseline.

Throughout this process, a set of working software documentation is created which, in fact, bounds the technical activity. Each step of the effort being with a formal specification and concludes with a specification defining the effort to be performed subsequently. Once approved, the documents produced at identifiable milestones during and at the conclusion of the software development process are used as a primary basis for management control and permit controlled communications both internally with the software activities and externally between the software activity and all other system development organizations.

If the computer program is being developed within the context of a large software (or combined hardware/software) system, explicit attention is paid to recognizing and establishing interfaces and making provisions for their control. The software management plan provides for Interface Specifications to do this. The software management plan further provides fragmentation — in a structured way — for the total package of software for purposes first of developing such fragments and later of testing and integrating them in building up a well-tested and validated total software package for release or delivery.

The software development, testing, integration, and documentation process, which is described here, is applied to both individual "standard" program units as well as to the development of large, complex software systems.

4.4.1.2 Specification

As shown in Fig. 4-6, Data Processing System Requirements (DPSR's) are based on applicable development specifications and other requirements source documents. The DPSR's state the requirements to which software will be developed, and form the software portion of the allocated baseline. DPSR's are followed by the Functional Specification. Functional Specification permits the segmentation of the computer program system into modular functions. A more complete description of the utilization of these and subsequent documents is contained in the Section Software Documentation Plan.

A Requirements Review is held for each DPSR. A Functional Design Review is conducted for each Functional Specification. SAM-D Project Office (AMC) Personel are invited to attend both Requirements and Functional Design Reviews. Both DSPR's and Functional Specifications are controlled by the Engineering Review/Engineering Change Order (ER/ECO).

4.4.2 Software Design

The horizontal scale of Fig. 4-6 indicates (approximately) the relative time phasing among the functional steps. The Functional

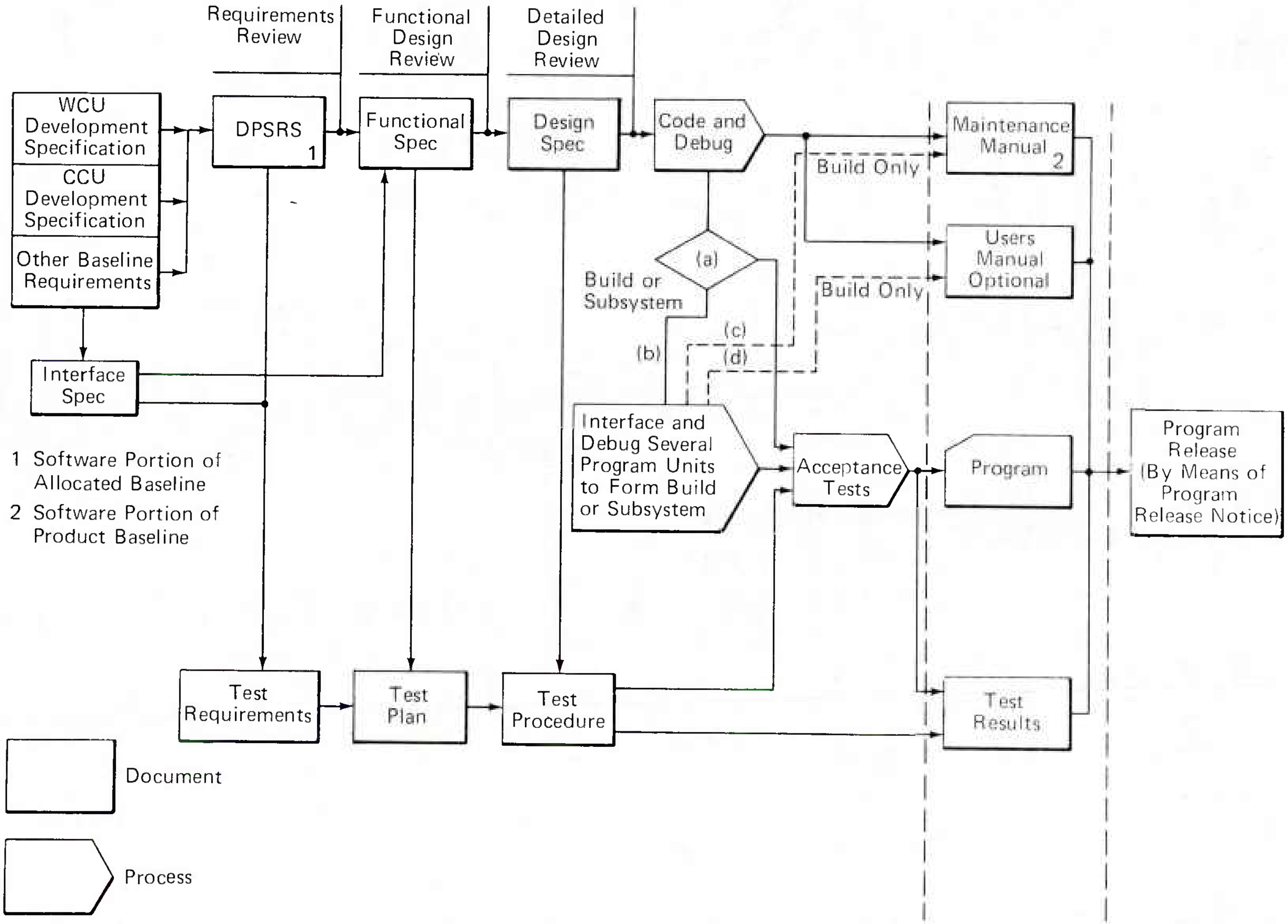


Fig. 4-6 Overall Computer Program Development Process

Specification and the Test Requirements occur in approximately the same time period following the production of DPSR's. The Test Requirements document must identify all system level tests for each build (although details of such tests are to be determined later) to permit adequate planning for the integration effort. The Design Specification and the Test Plan follow again in about the same time period. Formal SAM-D Project Office (AMC) participation is not required for a design review procedure for the Design Specifications.

4.4.3 Software Implementation

4.4.3.1 Code/Debug/Acceptance Test

Following approval of the program unit Design Specification, the program unit is coded and debugged by the programmer. After the programmer is satisfied that the program is debugged, acceptance testing is performed according to a previously written Acceptance Test Procedure.

Acceptance Tests:

The Acceptance Test is the formal qualification of a software unit, whether it be a program unit, subsystem, or build. The test follows the detailed procedure defined in the corresponding Test Procedure, and is deemed successful when the test results are reviewed with regard to expected results and accepted by appropriate authorities.

4.4.3.2 Release

Following successful completion of acceptance testing and publication of the program's Maintenance Manual, Users Manual (if required), and Test Results, the program may be released. The release process involves assurance that:

1. All required documentation has been published,
2. All required program media (card decks and tapes) are under control, and
3. Proper levels of supervision concur via the release.

The release process for either program units or system/subsystem builds will be accomplished by means of a release notice.

This software release procedure forces standardization of source card deck sequencing and insures that current documentation is available. Software modification via the ER/ECO procedure requires that all program changes are incorporated into the source deck and the documentation.

4.4.3.3 Versions/Modifications

This plan recognizes the need for "preliminary" or "informal" release of program units, as in cases where a program may be required for the scheduled integration or test of a build or subsystem before the formal release process has been accomplished. Thus, an identification procedure has been formulated wherein Program Units and Systems are identified by Versions and Modification identification at release.

4.5 SOFTWARE VALIDATION AND INTEGRATION

The software testing process is effected and controlled through a logical series of steps and documents. This process is discussed in the sections that follow.

4.5.1 Test Requirements

The purpose of the Test Requirements is to define the performance requirements for each test, each test itself, and its scope to assure the item will operate satisfactorily in the system. The Test Requirement identifies the functions to be tested and specifies the number of cases, ranges, and limits of data and hardware environment. If necessary, the complete requirements for all tests may be omitted from the first issue of a Test Requirement document to permit its publication in phase with an associated DPSR in order to permit adequate planning and scheduling for related software testing. Any such "to be determined later" data items shall be identified and the final document published in a timely manner. The document will be published a minimum of 30 days prior to the test, or as reflected by PERT.

4.5.2 Test Planning

This process is responsive to the requirements stated in the Software Test Requirements. Its purpose is to identify how and where each stated requirement is to be satisfied and to assure that the necessary test drivers, equipment configuration and availability, required program units,, and test data are identified. A Test Plan then identifies and documents the complete build up sequence required to test a software system, defines one or more tests, and associates each function to be tested with the relevant test or tests. For each test thus defined, a Test Procedure will be produced. The test plan also defines the testing required for the individual program unit components or subsystems.

In the development of stand-alone software, it may be appropriate to combine the Test Plan and Test Procedures which are described in the following paragraph, for issue as a single document.

4.5.3 Test Procedures and Results

4.5.3.1 Test Procedures

The Test Procedures describes the function or functions to be tested; describes the required program unit(s), test driver and/or equipment, components and any modification to their normal operating procedure required for purposes of the test; and provides the complete test data. It also provides the detailed, stepwise procedure required to run the test, and defines the expected results.

4.5.3.2 Test Results

The Test Results are intended to state how well the actual and expected results agree, and explain any differences. They also provide a complete record of the particular test for future reproduction. A Test Result document restates the test objectives, equipment configuration and references the Test Procedures document for other relevant information. A conclusions-and-recommendations section summarizes all efforts engendered by that test, and clearly indicates that the program implementation has been validated, i.e., all required functions have been satisfactorily demonstrated. The detailed results are contained in the document appendices.

4.5.3.3 Release

The Release, which is the formal completion of program unit, subsystem, or build acceptance testing, is accomplished when the test objectives defined in the Test Results document have been met, appropriate sequenced source program decks and/or tapes and compilation listings are filed, and the relevant approved documentation is published.

4.5.4 Simulation Support System Software

A Simulation Support System (S^3) will be designed and implemented to serve two functions in the Engineering Development software development effort:

1. Provision of software test capability in support of the Weapon Control Unit software system build effort,

2. Provision of a statistically valid software test tool for the demonstration of Engineering Development Fire Control Group performance capability.

S^3 use in support of software build testing occurs in the controlled environment of the Tactical Software Development Computer Facility. The environment model provides the capability of specifying the scenario required for a test, while the radar model provides simulated radar responses (in accordance with the specified scenario) to application program requests for search, track, and/or guidance action. S^3 can demonstrate WCU performance by applying simulation of quantities and types of targets that are not economically feasible for live testing of the system.

4.5.5 Support Software

The development of SAM-D Engineering Development application software establishes implicit requirements for software to support the development effort. This software used in the software development process to generate, operate, and verify the operational software being developed is categorized as operating systems and support software and is required for each of the computers in the Tactical Software Development Facility, including the Univac 1108, a nontactical WCC, and a tactical WCC.

4.5.5.1 Operating Systems

Each of the three Tactical Software Development Facility computer configurations requires an operating system to provide a complete, dynamic, and functional interface between the computer's users and the processing required by them. Each of these operating systems consists of three subsystems: a supervisor, which provides overall control through job or task sequencing; input/output modules, which control peripheral devices; and debugging modules, which assist the programmer in the checkout of software.

4.5.5.2 Support Software

Support software includes a compiler and its related software, assemblers, a digital simulator, various utility modules, and test support software.

The Engineering Development compiler is tailored to SAM-D software implementation needs and is designed to reduce software development lead time and facilitate software checkout. The compiler language chosen for SAM-D Engineering Development is JOVIAL, the same language used in Advanced Development. Compiler support software is also required to increase the effectiveness and utility of the compiler as an implementation tool.

The WCC assembler is used to generate machine dependent software, and will be used to produce parts of the diagnostics, operating systems, and executive software. The digital simulator provides a software verification tool independent of the hardware and is used to provide timely development in a completely controllable environment. The test support software is necessary to provide for the generation of test data and the recording and reduction of test results.

4.5.6 Acceptance Test Software

Special purpose computer programs designated Acceptance Test Software are required to verify that the Engineering Development Weapon Control Unit hardware meets the performance requirements designated in the Development Specification.

The programs comprising the acceptance test software system will be independent and self-contained in that no program requires a direct interface with any other program. However, they are dependent in the sense that it is necessary to define a sequence of testing in order to establish consistent design and implementation ground rules. Essentially, the approach to the Design Test Software is to first develop a computational capability utilizing self-test techniques and then use this capability as a test vehicle for testing the rest of the unit.

Briefly, the WCC is tested and then used to control and evaluate the performance of the Display and Communications equipments. The WCU systems test is used to establish that all the WCU sub-units operate as a unit while the final stage of an acceptance test consists of verifying the WCU interface signals to other equipments. The functional test maintains a WCU test capability after acceptance testing has been completed.

4.5.7 Test and Diagnostic Software

The SAM-D system requirements indicate a need for test software to support hardware maintenance and test. Test software is also required to support engineering and manufacturing checkout of the WCC as well as hardware interface verification.

The test and diagnostic software will be composed of the following major elements:

1. Fault detection software, required to establish that the WCC and peripheral equipment is operational. Test results will be simply go/no-go indications, with the failing unit identified if possible.

2. Fault isolation software, required to locate the fault to the basic replacement unit and to verify the operation of replaced units.

3. Interface verification software, required to verify all data and control paths between system units.

4.6 SOFTWARE ACQUISITION MANAGEMENT ORGANIZATION AND METHODS

4.6.1 General Management Information

SAM-D management information is sumarized in Table 4-4.

TABLE 4-4

SAM-D MANAGEMENT INFORMATION

Program Status	Recycled back to Advanced Development from Engineering Development
Program Manager	U.S. Army Material Command, Redstone Arsenal
System Contractor	Raytheon Missile Systems Division, Bedford, Mass.
Contract	Cost plus incentive fee
Software Contracts	Raytheon
Validation Agent	U.S. Army TECOM
Maintenance Agent	U.S. Army
Software Deliverables	User's manual, maintenance manual, and program tapes
Integration Agent	Raytheon

The SAM-D Project Office is organized along functional lines as shown in Fig. 4-7.

4.6.2 Management Planning

Table 4-5 depicts the key elements of the SAM-D Management Plan.

The major control will be the cost plus incentive fee contract and an award fee for meeting production major item unit cost goals.

A fully implemented Cost/Schedule Control System is employed, supported by the following:

1. A rational Work Breakdown Structure for allocation of requirements and collection of expenditures.

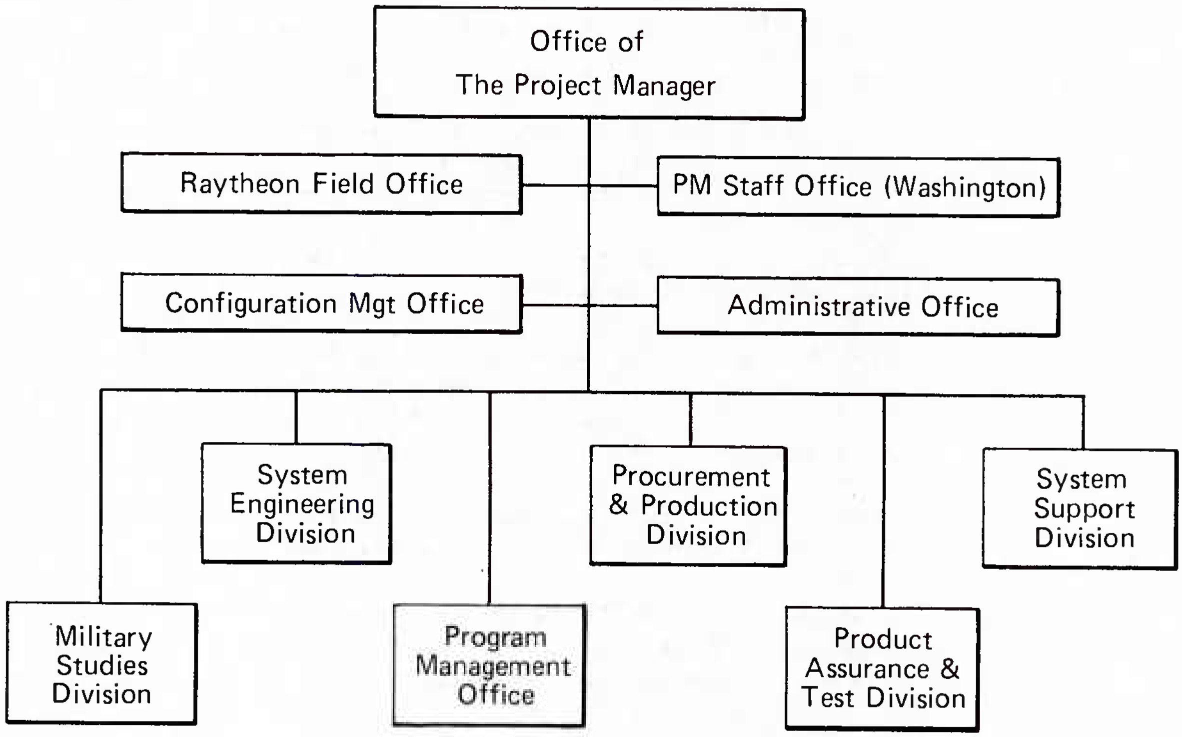


Fig. 4-7 Project Manager SAM-D Missile System, USAMC

TABLE 4-5

SAM-D MANAGEMENT PLAN: KEY ELEMENTS

Project Office	Functional Organization
Control	Cost Plus Incentive Fee Contract o Cost/Schedule Control System o System Specification o "Gating" Milestones o System Engineering Management Plan o 12 Other Specialty Plans
Overview	Quarterly Management Reviews
Assistance	DA Requirements Control Board System Engineering Cost Reduction Assistance Contractor

2. A System Specification which assures that top-level Material Need parameters are met.

3. A System Engineering Management Plan written by the contractor and approved by the Government that integrates a description of all system activities in one document for top management overview and control.

4. A series of "Gating" Milestones that focus management attention on the overall program development.

5. Twelve other specialty plans that spell out in detail the method and approach of the development program in such areas as environmental testing, reliability, maintainability, human engineering, etc.

In order to establish assurance of program progress, Quarterly Management Reviews will be held with the contractor. This assistance is provided by Requirements Control Board (RCB) and System Engineering Cost Reduction Assistance Contractor (SECRAC). The establishment of the Requirements Control Board is one of a number of related actions incorporated into the SAM-D Program to help reduce production costs and related life cycle/program costs. Proposed major requirement changes that have a significant impact on cost, schedule, and performance are considered by the Board.

The SAM-D Project Manager will review all proposed requirement changes for their proposed impact on program cost, schedule, and performance, and he will determine which ones should warrant consideration by the Board. Changes will be made specifically for cost reduction as well as for technical reasons.

From an operational concept point of view, the SAM-D Project Manager retains the authority to approve any action that will:

1. Affect the item's capability to meet requirements as specified in the System Specification and subsequent revisions thereto,

2. Increase costs, or

3. Affect the overall SAM-D Program Schedule.

Within industry, the SAM-D Project Manager has fourteen contracts with various firms and institutions, the majority of which are shown on Fig. 4-8. The prime contractor, Raytheon, has subcontracted with Martin Marietta at Orlando for the missile (less guidance and warhead) and the launcher. Martin in turn has subcontracted with Thiokol for the rocket motor.

To encourage the prime contractor, Raytheon, to develop and design SAM-D major items to a reasonable production unit cost, IBM was selected as the SECRAC. IBM is given specific Technical Direction Orders to examine cost reduction aspects of the SAM-D design.

Within Raytheon, responsibilities and associated procedures for ensuring the successful completion of the SAM-D software activities will be allocated between Program Office and Laboratories in the following manner.

Customer Liaison:

The Program Office is responsible for all customer liaison, and any supporting department will provide assistance at the Program Office's request.

Work Authorization Directive:

The Program Office will promulgate a work breakdown structure that meets customer and Program Office needs. The Program will then issue Work Authorization Directives that will contain the general requirements for the work to be performed.

Task Descriptions:

The performing organization (usually the Lead Engineer) will prepare detailed task descriptions in response to the Work Authorization

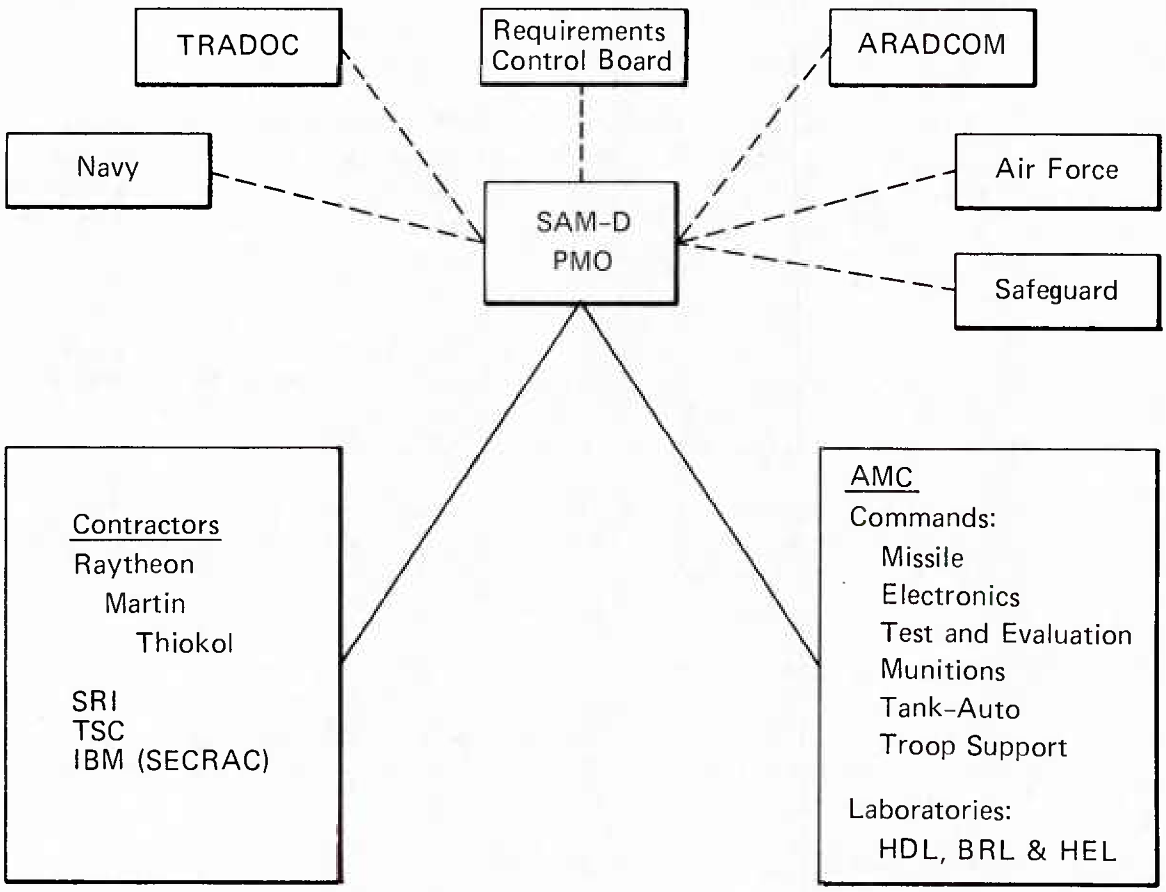


Fig. 4-8 SAM-D Participating/Interfacing Organization

Directives. These detailed task descriptions together with cost data and summary level PERT schedules will form the initial software development plan, which will be submitted to the Program Office for review and approval.

4.6.3 Software Subcontract Management

In addition to the applicable ASPR regulations and Raytheon Company's general policies covering subcontract activities, software subcontract management during the SAM-D engineering Development activity will be governed by the following:

1. No software tasks will be subcontracted outside of Raytheon without prior consideration being given to the use of available Raytheon resources.

2. Requests for subcontractor quotations will be approved by the SAM-D PMO prior to processing by procurement personnel in accordance with Missile Systems Division and Bedford Laboratory policies. Approval will be contingent upon review of a detailed statement of work, schedule requirements, contract award criteria for competitive procurements, and sole source justification (if applicable).

3. Subcontracts will be awarded on a competitive, fixed price basis, unless the task to be performed cannot be sufficiently well defined to satisfy fixed price procurement requirements, or if sole source procurement is clearly in order.

4. In those cases where time and materials type subcontracts are indicated, they will, in general, be performed on site, and not at the vendor's facility.

5. Acceptable performance will be specified with test cases.

6. All software subcontracting efforts will comply with the appropriate guidelines established in the Engineering Development Software Management Plan.

7. Key personnel retention and personnel selection approval will reside with Raytheon.

8. Subcontractors will be bound by the same documentation requirements and procedures as govern the Raytheon SAM-D software development effort, unless subcontractor standard procedures are deemed adequate.

9. Technical and management visibility will be assured through development and approval of a Program Plan (Software Development Plan) for each major software subcontract. The plan will detail major milestones, check points, schedules, etc.

10. Responsibility for monitoring and validating subcontractor activities, procedures, etc. will be a joint responsibility of the Laboratory Lead Engineer and Program Management Office Project Manager, as applicable.

4.6.4 Schedule and Cost Control

4.6.4.1 Schedule Control

SAM-D schedule control is maintained via the Program Evaluation and Review Technique (PERT). PERT networks provide graphically the planned sequence in which activities and events occur, as well as their interrelationships and interdependencies. They serve as the basis for estimating time, scheduling work, establishing priorities, determining costs, and communicating the work effort to all management levels.

The Program Office will establish overall delivery requirements for the subsystems/tasks assigned. Upon completion of negotiations between the Program Office and the performing organization relative to detailed task descriptions, milestones, and cost estimates, a summary level PERT is completed and will become the official schedule for the tasks assigned. Development and construction of the summary level networks will be in compliance with the standards established in the SAM-D PERT manual.

In addition to the summary level PERT network, detailed PERT networks for tasks assigned will be developed as required.

4.6.4.2 Cost Control

The prime factor in cost control is timely comparison between budgets and actual expenditures. Software financial control is the responsibility of SAM-D Program Management Office and will conform to the requirements specified in SAM-D Cost/Schedule Control System Plan.

In addition to these fiscal controls, software management will also comply with additional cost controls established by the performing organization.

Within the software development effort, costs will be established at a functional work package level. The budget will be based upon negotiations between the SAM-D Program Office and the performing organization using the Work Authorization Directive, detailed Task Descriptions, and schedules. The approved budget establishes the baseline for cost control.

4.6.5 Quality Assurance

Within the SAM-D Engineering Development Software development activity, quality assurance is essentially two-fold. It is a continuing process: (a) to demonstrate that the software at every level meets the performance criteria of the applicable specifications, and (b) to ensure a high level of consistency or efficiency, reliability, and maintainability.

The above objectives are achieved by the software development process. The software documentation provides visibility into the design and implementation activity with specific attention given to the test area; i.e., development and execution of test requirements, test plans, test procedures, and analysis and documentation of test results. The review activity with approval at key milestones provides assurance that quality objectives are met.

4.6.6 Configuration Management

Configuration management requirements and procedures for SAM-D computer programs and associated documentation will be implemented and maintained in accordance with the provisions of the SAM-D Configuration Management Plan.

Documentation for software will be prepared and controlled in accordance with the procedures outlined in the Configuration Management Plan.

The following documents, subject to ER/ECO control, shall be prepared in accordance with Form 3 (Specifications to Commercial Practices) of MIL-S-83490 and Contractor formats as identified within MIL-S-83490:

1. Data Processing Systems Requirements,
2. Interface Control Specifications,
3. Functional Specifications,
4. Test Requirements Specifications (Software),
5. Software Test Plans, and
6. Maintenance Manuals

Change control to be utilized by the contractor for SAM-D data processing system and software documentation will be as specified below. SAM-D Project Office change control of data processing system and software documentation is via ER/ECP control of DPSR's and Maintenance Manuals.

Maintenance Manuals (software portion of the product baseline) will be released to Government control subsequent to the establishment of the hardware portion of the product baseline.

ER/ECO control shall be exercised by the contractor on the six listed software document categories. Release and change control of software documentation shall be in accordance with the SAM-D Configuration Management Plan. The services of the Engineering Release services group, ECO coordinators, microfilm services and libraries, central vault, and reproduction facilities will be made available and utilized in conjunction with the Engineering Release System.

4.6.7 Software Documentation Plan

The SDP is tailored to the SAM-D Engineering Development software development effort.

This plan places emphasis on:

1. Explicit statement of requirements;

2. Documentation of interfaces;

3. Review and approval of design;

4. Planned, formal, controlled testing;

5. Complete documentation of final programs; and

6. Control of program change.

The SDP does not attempt to cope with all SAM-D software development efforts by applying a rigid set of documentation that is to be rigorously applied without considering the utility of and need for each type of document. Instead, the SDP establishes a basic set of documentation and a series of application guidelines that establish documentation requirements in a spirit of doing a systematic, complete, and professional job. These guidelines will be a subject for continuing review, and should be modified as required on a cost benefit basis — the dollar spent producing a document not required on a particular task, and the dollar spent in compensating for the lack of a necessary document that was not initially specified in the application guidelines are equally wasted. Not surprisingly, these wasted dollars have a concomitant effect on schedule, and result in an indirect cost penalty associated with delays in related schedules which, although difficult to qualify, cannot be ignored.

4.6.7.1 Data Processing System Requirements

The software development process will be based on an explicit set of requirements that are documented as the Data Processing System Requirements along with companion hardware requirements and combined interfaces as one of the results of the system engineering process. The purpose of the Data Processing System Requirements is to provide a complete, explicit statement of the requirements that the data processing system is to satisfy.

The Data Processing System Requirements will function as a complete expression of the requirements for the data processing system. The Data Processing System Requirements will be maintained and will function at all times as the software requirements baseline.

4.6.7.2 Interface Specifications

An Interface Specification will be created covering all interfaces between the subject software system and related hardware, and the software system and other software either existing or being developed. The purpose of the Interface Specification will be to describe essentially the environment in which the software system is to fit.

While it is not feasible to create a complete Interface Specification at the time that the Data Processing System Requirements is published, the Interface Specification should nevertheless be established at that point to indicate total scope and format. The incomplete sections of the Interface Specification should be the source of continuing effort and must be resolved prior to the completion of the design specification.

4.6.7.3 Functional Specifications

Following approval and release of the Data Processing System Requirements, a Functional Specification will be developed for the software system.

The purpose of the Functional Specification is to:

1. Document the overall software design approach,

2. Fully describe the manner in which the software will be structured,

3. Document all interfaces not included in Interface Specification, and

4. Describe where and how each requirement in the Data Processing System Requirements will be satisfied.

Normally a single Functional Specification will be produced for a software subsystem. In those cases where the system is complex or the development must be time phased to accommodate the availability of required information or interrelated development efforts, the Functional Specification may be developed as a tree structure of specifications. When the alternative of a Functional Specification tree is chosen the top level of detail will explicitly identify each lower level Functional Specification and the area it is to cover, and all interfaces between the subsystems.

4.6.7.4 Design Specifications

A Design Specification will be produced for each program unit or module. The purpose of the Design Specification is to document the complete design of a program unit before there is a commitment to coding so that the design can be reviewed to determine:

1. That all functions allocated to the program unit are covered;

2. That the algorithms employed are appropriate, and

3. That all interfaces will be met.

4.6.7.5 Maintenance Manual

A Maintenance Manual will be produced for each program unit. The purpose of the Maintenance Manual is to provide complete "as built" documentation for the program unit.

Contrary to hardware, the set of software documentation is subject to significant change during the "production" or coding process. The Maintenance Manual provides for this change by becoming the as built replacement for the Design Specification. Upon release of the Maintenance Manual for a program unit the corresponding Design Specification becomes obsolete. The Maintenance Manual includes the program listing which consists of both the source and corresponding object code; i.e., describes the conditions under which the listing was produced (i.e., COMPOOL identification, compiler, or assembler version, etc.).

4.6.7.6 Users Manual

For program units in which there is a user/program interface, a Users Manual is required. The users manual provides two types of information. First, it provides the step/procedures necessary to setup/initialize the system and identifies the minimum equipment configuration required. It covers such items as system generation constants, parameters, default values, input-output device assignments, etc. Second, the Users Manual covers the techniques/information/actions necessary when the software system is in operation. Subjects that are covered include indications via the operator's console/printouts that the system is operating correctly, diagnostic messages that are most likely to appear if any problems occur, etc.

4.6.7.7 Test Requirements

The Test Requirements document identifies the functions to be tested and specifies the number of cases, ranges, and limits of data and hardware environment. The Test Requirements document leads directly to the preparation of the Test Plan.

4.6.7.8 Test Plan

The Test Plan is a single document responsive to the requirements and to where each stated requirement is to be satisfied, and identifies the necessary test drivers, equipment configuration and availability, required program units, and test data. The Test Plan then identifies and documents the complete build up sequence required to test a software system, defines one or more tests, and associates each function to be tested with the relevant test or tests. For each test thus defined, a Test Procedure will be produced. The Test Plan also defines the testing required for the individual program unit components or subsystems.

In the development of some small stand-alone programs, it may be appropriate to combine the Test Plan and the Test Procedures, which are described in the following paragraph, for issue as a single document.

4.6.7.9 Test Procedures

The Test Procedures document describes the function or functions to be tested; describes the required program unit(s), test driver and/or equipment, components and any modification to their normal operating procedure required for purposes of the test; and provides the complete test data. It also provides the detailed, stepwise procedure required to run the test, and defines the expected results.

4.6.7.10 Test Results

The Test Results Document states how well the actual and expected results agree, and explains any differences. It also provides a complete record of the particular test for future reproduction, documents the satisfactory completion of all test objectives and equipment configuration used, and references the Test Procedures document for other relevant information. The Conclusions and Recommendations sections summarize all efforts engendered by the test, and clearly indicates that the program implementation has been validated; i.e., all required functions have been satisfactorily demonstrated. The detailed results are contained in the document appendices.

4.6.8 Documentation Revisions

For the purpose of clarity, the documents in Fig. 4-6 are shown in a straight-line flow without indication of the need for rework or revision. However, it must be noted that changes in the documents shown in this figure may propagate changes throughout all completed code and documentation (i.e., functional/design specifications, maintenance/users manuals, and acceptance test procedures and results). During software test and use, requests for program modifications and/or software problem reports will also require software changes. All changes are implemented under control procedures that insure that the revised documentation will track the modified code. Thus, every documentation change to a program can be chronologically traced from the current released version/modification back to the original released version/modification. Each program release will identify the corresponding documentation, symbolic card deck, listing, object magnetic tape, and data base (COMPOOL) used.

Each performing organization (usually Lead Engineer) will also be required to maintain records of all requests for program modification and software problem reports and their disposition.

4.7 OPERATIONAL SOFTWARE MAINTENANCE

After deployment of the SAM-D system, a maintenance facility will be employed at the Redstone Arsenal under the U.S. Army. As part of the documentation requirement for the software, a maintenance manual will be available. The maintenance manual consists of the software design specifications plus program listings.

Little formal documentation is available on maintenance plans for SAM-D. The program manager states that program modularization together with adequate program documentation should keep maintenance costs down after system production and deployment.

4.8 HIGHLIGHTS

Prior to full-scale development, the software requirements for the major SAM-D functions were delineated in a set of over 150 documents called the Data Processing System Requirements (DPSR's) prepared by the contractor. The DPSR's are reviewed by the Program Manager Office. Core and timing estimates are also made, but these have much uncertainty. (MP1)

The PERT system of program scheduling has been used. The SAM-D Software Management Plan prepared by the contractor specifies in detail the milestones for software development covering analysis, design, implementation, integration, and testing. (AP1, AP2)

Extensive system tradeoff analyses were conducted by the contractor during the Engineering Development Definition Phase of SAM-D. The analyses included a study to determine the system functions allocated to the dedicated hardware of the radar unit and to the general purpose computer in the weapon control unit. (SE1)

The development of the software for the SAM-D system was identified as a high risk item during the advanced development phase. Provisions were made for expansion of the memory and CPU capability in the event that timing and sizing estimates for the software were inadequate. A top-down approach is employed in software development documents. (SE2)

The software and unique computer are undergoing parallel development at the prime contractor's facility. JOVIAL higher order language is being used to develop the computer instructions. For software checkout purposes, the SAM-D computer is emulated on a Univac 1108. (SE3, IP1)

As part of the software support tools for the SAM-D system, the contractor developed the following: a JOVIAL compiler and related assembler, various utility modules, data base, and operating system. The Program Manager Office has had difficulty in monitoring the software development because of the use of these nonstandard support tools and the use of a modified form of JOVIAL as a higher level programming language. (IP1)

Software integration is accomplished in the SAM-D Tactical Software Development Facility. The contractor and Program Manager felt that this type of software test-bed facility should be encouraged for application to other programs by appropriate funding.

Prototype and tactical software adequacy is verified with a test-bed simulator at the prime contractor's facility, followed by operational integration assessment at White Sands. (IP3)

The SAM-D Project Office has a staff of eight software specialists to monitor progress and to approve the prime contractor's plans with respect to software development. The Project Office has been assisted by IBM Federal Systems Division, which has provided an independent assessment of certain areas of the software development. (MS1)

INITIAL DISTRIBUTION EXTERNAL TO THE APPLIED PHYSICS LABORATORY*

The work reported in SR 75-3D was done under Navy Contract N00017-72-C-4401. This work is related to Task ZC-6, which is supported by NAVSEASYSCOM.

ORGANIZATION	LOCATION	ATTENTION	No. of Copies
DEPARTMENT OF DEFENSE			
DoD Software Management Steering Committee	OASD (I&L), Pentagon Room 2A318	B. C. DeRoze	25
<u>Department of the Navy</u>			
Naval Material Command	NC-3 2531 Jefferson Davis Hwy., Arlington VA 20362	Trident Proj Mgr. CNM-PMZ	2
<u>Department of the Army</u>			
MICOM (Army Missile Command)	Redstone Arsenal Huntsville, AL 35809	Pershing Program Mgr.	2
Army Material Command	Redstone Arsenal Huntsville, AL 35809	SAM-D Program Mgr.	2

Requests for copies of this report from DoD activities and contractors should be directed to DDC, Cameron Station, Alexandria, Virginia 22314 using DDC Form 1 and, if necessary, DDC Form 55.

*Initial distribution of this document within the Applied Physics Laboratory has been made in accordance with a list on file in the APL Technical Publications Group.